

20
20

GELECEĞİN DÜNYASINDA BİLİMSEL VE MESLEKİ ÇALIŞMALAR

BİLGİSAYAR MÜHENDİSLİĞİ / I

EDİTÖRLER

DR. ÖĞR. ÜYESİ EMRE AVUÇLU

DR. ÖĞR. ÜYESİ DURSUN EKMEKÇİ

EKİN 
Basım Yayın Dağıtım

**GELECEĞİN DÜNYASINDA
BİLİMSEL VE MESLEKİ ÇALIŞMALAR
2020
BİLGİSAYAR MÜHENDİSLİĞİ / I**

EDİTÖRLER

DR. ÖĞR. ÜYESİ EMRE AVUÇLU
DR. ÖĞR. ÜYESİ DURSUN EKMEKÇİ

MART 2020
BURSA / TÜRKİYE

EKİN 
Basım Yayın Dağıtım

Editörler

DR. ÖĞR. ÜYESİ EMRE AVUÇLU
DR. ÖĞR. ÜYESİ DURSUN EKMEKÇİ

Birinci Baskı • © Mart 2020 / Bursa

ISBN • 978-625-7983-95-2

© copyright

All Rights Reserved

Ekin Basım Yayın Dağıtım

Tel: 0224 223 04 37

Mail: info@ekinyayinevi.com

Web: www.ekinyayinevi.com

Adres: Şhreküstü Mahallesi Cumhuriyet Caddesi
Durak Sokak No:2 Osmangazi - Bursa

EKİN 
Basım Yayın Dağıtım

İÇİNDEKİLER

İÇİNDEKİLER.....	i
POPÜLASYON TABANLI META-SEZGİSEL OPTİMİZASYON TEKNİKLERİ.....	1
1. GİRİŞ.....	1
2. META-SEZGİSEL OPTİMİZASYON ALGORİTMALARI.....	2
2.1. Genetik Algoritma.....	3
2.2. Parçacık Sürüsü Optimizasyonu.....	6
2.3. Karınca Koloni Optimizasyonu.....	8
2.4. Yapay Arı Kolonisi Algoritması.....	12
3. SONUÇLAR.....	14
4. KAYNAKÇA.....	14
FEROMONAL YAPAY ARI KOLONİ ALGORİTMASI (fYAK).....	17
1. GİRİŞ.....	17
2. YAPAY ARI KOLONİ ALGORİTMASI.....	18
2.1. Bal Arısı Kolonisinin Yiyecek Toplama Sürecindeki Kolektif Davranışı.....	18
2.2. YAK Algoritması Ana Modülleri.....	19
2.3. Algoritma Adımları ve Akış Diyagramı.....	21
3. KARINCA KOLONİ OPTİMİZASYONU.....	23
3.1. Karınca Koloni Sistemi.....	24
4. FEROMONAL YAPAY ARI KOLONİ (fYAK) ALGORİTMASI.....	25
4.1. Kombinatoriyal Optimizasyon Problem Çözümleri İçin fYAK Modeli.....	26
4.2. Sayısal Optimizasyon Problem Çözümleri İçin fYAK Modeli.....	29
5. SONUÇLAR.....	32
6. KAYNAKÇA.....	32
MAKİNE ÖĞRENME ALGORİTMALARINA GENEL BAKIŞ.....	34
1. GİRİŞ.....	34
2. MAKİNE ÖĞRENME TERMİNOLOJİSİ.....	35
2.1. Model.....	35
2.2. Özellik (Feature).....	35
2.3. Hedef (Target).....	35
2.4. Eğitim (Training).....	35
2.5. Tahmin (Prediction).....	36
3. MAKİNE ÖĞRENME YÖNTEMLERİ.....	36
3.1. Danışmanlı Öğrenme (Supervised Learning).....	36
3.2. Danışmansız Öğrenme (Unsupervised Learning).....	36
3.3. Yarı Danışmanlı Öğrenme (Semi-supervised Learning).....	36

3.4. Takviyeli Öğrenme (Reinforcement Learning)	36
4. MAKİNE ÖĞRENME PROBLEMLERİ	37
4.1. Sınıflandırma (Classification).....	37
4.2. Regresyon (Regression)	37
4.3. Kümeleme (Clustering).....	37
4.4. Yoğunluk Tahmini (Density Estimation).....	37
4.5. Boyut İndirgeme (Dimensionality Reduction)	38
4.6. Sıralama.....	38
5. MAKİNE ÖĞRENME ALGORİTMALARI	38
5.1. Decision Tree.....	38
5.2. Random Forest.....	40
5.3. K-Nearest Neighbor (k-NN).....	41
5.4. Support Vector Machine (SVM).....	42
5.5. Naive Bayes.....	46
5.6. Multinomial Logistic Regression.....	47
6. SONUÇLAR	48
7. KAYNAKÇA	49
KURUMSAL BİLGİ SİSTEMİ UYGULAMALARI İÇİN POPÜLER PROJE	
YÖNETİM YAKLAŞIMLARI.....	51
1. GİRİŞ.....	51
2. PROJE YÖNETİMİ.....	52
2.1. Proje Nedir?.....	52
2.2. Proje Yönetimi.....	53
2.3. Yazılım Proje Yönetimi	54
3. PROJE YÖNETİM YAKLAŞIMLARI	55
3.1. Çevik Yöntemler.....	55
3.2. Disiplinli Yöntemler	61
4. PROJE YÖNETİMİNDE BAŞARILI OLMAK İÇİN ÖNERİLER.....	66
5. KAYNAKÇA	66
MEME KANSERLERİNİN TEŞHİSİNDE KULLANILAN GÖRÜNTÜ İŞLEME	
TEKNİKLERİYLE İLGİLİ BİR DERLEME ÇALIŞMASI	69
1. GİRİŞ.....	69
2. MEME ANATOMİSİ.....	70
3. MEME KANSERİ VE İSTATİSTİKLER	71
4. MAMOGRAFİ	72
5. BİLGİSAYAR DESTEKLİ TEŞHİS SİSTEMLERİ.....	74
5.1. Önışleme Aşaması	74
5.2. Bölütleme Aşaması	76

5.3. Öznitelik Çıkarımı Aşaması	77
5.4. Sınıflandırma Aşaması	78
6. SONUÇ	80
7. KAYNAKÇA	80
BULUT SİSTEMLERİNE YAPILAN DAĞITILMIŞ HİZMET REDDİ SALDIRILARI VE ALINABİLECEK ÖNLEMLER.....	84
1. GİRİŞ	84
2. BULUT BİLİSİM SİSTEMLERİNİN YAPISI	86
3. BULUT SİSTEMLERİNE YAPILAN SALDIRILAR	87
4. BULUT SİSTEMLERİNE YAPILAN DDOS SALDIRILARININ TESPİT EDİLMESİ	88
5. BULUT SİSTEMLERİNE YAPILAN DDOS SALDIRILARININ ÖNLENMESİNDE KULLANILAN BAZI YÖNTEMLER	90
6. SONUÇ	92
7. KAYNAKÇA	93

POPÜLASYON TABANLI META-SEZGİSEL OPTİMİZASYON TEKNİKLERİ

Abdullah Elen*, Dursun Ekmekci

Karabük Üniversitesi T.O.B.B. Teknik Bilimler M.Y.O. Bilgisayar Teknolojileri Bölümü, Karabük.

aelen@karabuk.edu.tr

** Sorumlu Yazar*

1. GİRİŞ

Optimizasyon, bir problemin belirli koşullar altında istenen kısıtları sağlayan muhtemel en iyi çözümünü bulmak için özel olarak tasarlanmış yöntem ve teknikleri inceleyen bir matematik ve hesaplamalı bilim dalıdır. Optimizasyon, mühendislik, ticaret, ulaşım, finans, tıp ve herhangi bir karar verme süreci gibi alanlarda yaygın olarak uygulanmaktadır.

Doğrusal programlama, doğrusal olmayan programlama, dinamik programlama veya ayrık yapı (kombinatorial) gibi çok çeşitli problemleri çözmek için çok sayıda geleneksel optimizasyon tekniği tasarlanmıştır. Optimizasyon problemleri, çözüm bileşeni olarak nesnelere ya da gerçek değerler alabilen, bir veya daha fazla bağımlı değişkene dayanan, aynı zamanda bir veya daha fazla amacı en iyilemeye çalışan problemlerdir. Problem bileşenleri reel sayılardan meydana geliyorsa, problemin çözüm uzayında sonsuz sayıda çözüm olduğu anlamına gelir. Eğer bu bileşenler nesnelere oluşuyorsa, bu tür problemler kombinatorial optimizasyon problemleri olarak isimlendirilir. Kombinatorial optimizasyon problemlerinde, çözümler, çözüm bileşeni olan nesnelere sıralanmasıyla oluşur. Nesnelere, en iyi sonucu verecek biçimde sıralanışı, optimal çözümü oluşturur. Bu problem türlerinde, problem bileşenlerine eklenen her bir nesne, çözüm uzayını (ve bu bağlamda çözüm zamanını) giderek artırmakta ve kabul edilebilir zamanda çözülemeyecek boyutlara taşımaktadır. Dolayısıyla kombinatorial optimizasyon problemleri, kabul NP problemler sınıfında değerlendirilir. Eğer eklenen her bir problem nesnesi, bir önceki duruma kıyasla, çözüm zamanını daha çok artırmışsa, bu tür problemler NP-tam sınıfında, polinom zamanda çözümü olduğu ispatlanamayan seviyedeyse NP-zor sınıfına dahil edilir.

Günlük yaşamda çoğu kez farkında olmadan karşılaştığımız bu problemler, tek amaçlı ya da çoğu kez birbiriyle çelişen çok amaçlı problemler olabilir. Tek amaçlı optimizasyon problemlerinde, problemin yapısına göre amaç; minimum maliyet ya da maksimum fayda olabilmektedir. Çok amaçlı problemlerde ise amaçlar aynı yönlü olabildiği gibi zıt yönlü de olabilir. Örneğin tüketiciler ürün seçiminde, maksimum fayda ile minimum fiyat arasında optimal çözüm noktasını bulmaya çalışır. Bu problemler, kişiler için olduğu kadar, işletmeler için de özellikle çözüm aranan problemlerdir. Gerçek dünya problemlerine uygulanan mevcut geleneksel optimizasyon tekniklerinin birçoğu; yerel optimal çözümlerin aktarılmasında zorluklar, ıraksama riski, birinci veya ikinci dereceden türevlerin hesaplanmasına ilişkin kısıtlamaların ele alınmasındaki zorluklar veya sayısal zorluklar gibi problemler konusunda belirgin bir hassasiyetten mustarıdır. Bu sorunların üstesinden gelmek için 70'li yıllardan itibaren birçok sezgisel ve meta-sezgisel teknikler önerilmiştir (Gavrilas, 2010). Meta-sezgisel algoritmalar, her ne kadar optimum çözümü garanti etmeseler de makul çalışma sürelerinde geçerli çözümler üretebilirler. Kesin yöntemlerden farklı olarak bu yöntemler, basit ve kompakt bir teorik desteğe sahiptir ve genellikle ampirik doğa kriterlerine dayanır. Doğal yaşamda, yinelemeli olarak en iyiye ulaşmayı hedefleyen bir süreç ya da koloni halinde yaşayan hayvan veya mikroorganizmaların aralarındaki iş bölümü, bu algoritmaların geliştirilmesine ilham kaynağı olabilmektedir. Bu bağlamda genellikle popülasyon tabanlı meta-sezgisel yöntemler (PTMSY) olarak geliştirilen bu algoritmalar, birçok ortak kavramı paylaşır.

PTMSY’de ilk olarak, belirlenen sayıdaki çözümden oluşan bir başlangıç popülasyonu oluşturulur ve çözümlerin başarı seviyeleri ölçülür. Ardından bu çözümleri kullanarak, algoritmalar kendi özel parametreleriyle, nihayetinde optimal çözüme (en yüksek başarı seviyesine sahip çözüm) ulaşmayı hedefleyen, daha başarılı yeni çözümler türetir. Bu algoritmaların zeki algoritmalar olarak tanımlanmasının sebebi de burada yatmaktadır. Genel olarak her bir çevrimde, başlangıçta belirlenen popülasyon sayısı kadar başarılı çözümle işlemlere devam edilir. Üretim ve değiştirme aşamaları hafızasız olabilir. Bu durumda, iki prosedür sadece mevcut nüfusa dayanmaktadır. Aksi takdirde, bir bellekte saklanan aramanın bazı geçmişi, yeni nüfusun üretilmesinde ve eski nüfusun değiştirilmesinde kullanılabilir. Yinelemeli olarak devam eden bu işlemler durdurma ölçütü (uygunluk kriteri) sağlanana kadar devam eder. PTMSY’in yaygın bir kullanım alanına sahip popüler örnekleri; Genetik Algoritma (GA), Parçacık Sürü Optimizasyonu (PSO), Karınca Koloni Optimizasyonu (KKO) ve Yapay Arı Koloni (YAK) Algoritması olarak sıralanabilir (Talbi, 2009).

Bu çalışma, optimizasyon problemlerinin çözümünde kullanılan popülasyon tabanlı meta-sezgisel tekniklere genel bir bakış sunmaktadır. Örnek olarak incelediğimiz yöntemler sırasıyla; GA, PSO, KKO ve YAK’tır. Çalışmanın içeriğinde; bu yöntemlerde kullanılan terimler, esin kaynakları ve genel olarak çalışma prensiplerinden bahsedilmektedir.

2. META-SEZGİSEL OPTİMİZASYON ALGORİTMALARI

Meta-sezgisel terimi, bir problemin sınırlarını belirleyen arama uzayında daha iyi sonuçlar elde etmek için diğer sezgisel yöntemlere rehberlik etmek amacıyla kullanılan yüksek seviyeli sezgisel bir terimin tanımlanması için önerilmiştir (Glover, 1986). Bu bağlamda yeni paradigmlar meta-sezgisel olarak adlandırıldı ve ilk olarak 80’li yılların ortalarında, birkaç genel sezgisel tarama seti kullanarak karmaşık optimizasyon problemlerine uygulandı. Bu yöntemler, problemin optimal çözümlerine yaklaşabilen ve hatta çözebilen bir arama algoritmaları ailesi olarak tanıtıldı. Her ne kadar geleneksel stokastik arama yöntemleri esasen tesadüfen yönlendirilirse de (çözümler bir adımdan diğerine rastgele değişir), arama sürecini yönlendirmek ve yakınsamayı hızlandırmak için meta-sezgisel algoritmalarla birlikte kullanılabilirler. Çoğu meta-sezgisel algoritma her zaman global optimum çözümü bulamaz ve bu nedenle elde edilen sonuçlar yalnızca yaklaşık değerlerdir, yani kesin değildir. Ancak bir meta-sezgisel yöntemin en çekici özelliği, çözülmesi istenen optimizasyon problemi hakkında özel bir bilgiye ihtiyaç duymamasıdır. Bu nedenle optimizasyon problemleri veya diğer türdeki problemler için genel bir çözüm modeli kavramını tanımlamak için kullanılabilir (Blum & Rolli, 2003).

Bugüne kadar birçok bilim insanı yaptıkları çalışmalarda, optimizasyon problemlerini çözmek için meta-sezgisel yöntemleri sürekli olarak geliştirdi. Böylece daha önceleri çözülmesi zor ve hatta imkânsız olduğu düşünülen problemlerin çözümüne olanak sağlandı. Yaygın kullanım alanına sahip optimizasyon algoritmaları olarak: Tavlama Benzetimi (TB), Tabu Arama (TA), Evrimsel Algoritma (EA), Yapay Bağışıklık Sistemi (YBS), Memetik Algoritmalar (MA), Parçacık Sürü Optimizasyonu (PSO), Karınca Koloni Optimizasyonu (KKO), Diferansiyel Evrim (DE), Uyum Arama (UA), Yapay Arı Koloni (YAK) vb. algoritmaları sayılabilir.

Meta-sezgisel yöntemler son yıllarda giderek güç kazanmakta ve karmaşık optimizasyon problemlerinin çözümünde sıklıkla tercih edilmektedir. Bunların nedeni aşağıdaki gibi özetlenebilir (Altunbey ve Alataş, 2015):

- Eş zamanlı olarak, farklı tipte karar değişkenleri, amaç fonksiyonu ve kısıtlayıcı fonksiyonları olması durumunda probleme uygulanabilecek genel çözüm stratejileri sunmaktadırlar.
- Çözüm stratejileri amaç fonksiyonu ve kısıtlayıcı fonksiyonların tipine ve problemi modellemede kullanılan değişkenlerin tipine bağlı değildir.
- Çözüm uzayı tipine, karar değişken sayısına ve kısıtlayıcı fonksiyon sayısına bağlı değildir.
- Sistemin modeli ve amaç fonksiyonu için kurulması zor olan ve bazen de kurulup da çözüm zamanı maliyeti çok yüksek olduğundan kullanılmayan çok iyi tanımlanmış matematiksel modellere ihtiyaç duymamaktadır.

- Hesaplama güçleri iyidir yani aşırı derecede hesaplama zamanına ihtiyaç duymazlar.
- Dönüşümleri ve adaptasyonları kolaydır.
- Klasik algoritmalarındaki gibi verilen bir probleme bir çözüm algoritması uyarlamada geçerliliğinin onaylanması zor olabilen bazı varsayımları gerektirmemektedir.
- Klasik algoritmalarındaki gibi ilgilenilen problem üzerinde değişiklik gerektirmemektedir. Farklı türdeki problemleri çözmek için kendilerini uyarlarlar.

2.1. Genetik Algoritma

Genetik algoritmaların (GA) temeli Darwin'in doğal seçim ve evrim ilkelerine dayanır. Bu anlamda GA oluşturulan popülasyonda bulunan ve bu popülasyonun algoritmasında bulunan bireylerden güçlü olanlar kullanıp yeni ve daha güçlü bireyler topluluğunu oluşturmaktır. Toplulukta en iyi bireyler hayatta kalır ve iyi bireylerin özelliklerini taşıyan daha iyi bireyler oluşturulması hedeflenmektedir. Genetik algoritmalar; problem için tek bir çözüm üretmeden ziyade birden fazla çözüm üreterek daha geniş bir çözüm havuzu oluşturarak daha etkin bir sonuç elde etmektedir.

Evrimsel Algoritmalar olarak ifade edilen GA, doğadaki evrimsel süreçleri model olarak kullanan bilgisayara dayalı problem çözme teknikleridir. Geleneksel programlama teknikleri ile çözülmesi güç olan, özellikle sınıflandırma ve çok boyutlu optimizasyon problemleri daha kolay ve hızlı olarak çözülebilmektedir. İlk defa 1960'larda psikoloji ve bilgisayar bilimi uzmanı olan John Holland tarafından geliştirilen GA, birçok bilimsel alanda genişçe işlenmiş, deneyleri yapılmış ve başarılı bir şekilde uygulanmıştır (Li ve diğ., 2009). Mekanik öğrenme konusunda çalışan Holland, Darwin'in evrim kuramında etkilenecek canlılarda yaşanan genetik süreci bilgisayar ortamında aktararak çeşitli sonuçlar almayı hedeflemiştir. GA'lar karmaşık çok boyutlu arama uzayında en iyinin hayatta kalması ilkesine göre bütünsel en iyi çözümü arar. GA'yı diğer arama yöntemlerinden (tabu arama, benzetilmiş tavlama gibi.) ayıran en belirgin özellikleri çözüm arama şeklinin farklı oluşudur. Aşağıda bu farklılıklar açıklanmaktadır (Goldberg, 1989).

- GA, parametrelerin kendisiyle değil, doğrudan parametre kodlarıyla uğraşır. Parametreler kodlanabildiği sürece çözüm üretir.
- GA tek bir noktadan değil, popülasyonun büyüklüğü kadar noktadan arama yapar.
- GA, ne yaptığını değil, nasıl yaptığını bilir. Yani GA önceden elde edilmiş bilgiyi değil, sadece amaç fonksiyonundan elde edilen bilgiyi kullanır.
- GA'nın uygulamasında kullanılan operatörler rastlantısal yöntemlere dayanır; belirli ve kesin yöntemler kullanmazlar.
- Genetik algoritmalar türev yerine hedef fonksiyonunun değerini kullanır.

Bunlara ek olarak GA, arama uzayının büyük ve karmaşık olduğu, mevcut bilgiyle sınırlı arama uzayında çözümün zor olduğu, problemin belirli bir matematiksel modelle ifade edilemediği, geleneksel eniyileme yöntemlerinden istenen sonucun alınmadığı alanlarda etkili ve kullanışlı bir yöntemdir. GA'da belirlenen kısıtlara uygun bir çözüme ulaşmak için amaç (uygunluk) fonksiyonunun problemin türüne göre tanımlanması, kromozom ve gen yapısının oluşturulması gerekir.

- **Gen:** Yapısında probleme ait en küçük bilgiyi taşıyan birime gen denir. GA'nın kullandığı programlama yapısında bu gen yapıları programcının tanımlaması ile ifade edilir. Bir genin yapısında ikili tabandaki sayıları, tamsayı, gerçel sayı, ağaç biçimini, farklı sembolik ifadeler gibi ifadeleri de içerebilir.
- **Kromozom:** Bir veya birden fazla gen yapısının bir araya gelerek problemin çözümüne ait bilgilerin bir kısmını oluşturan diziler kromozom olarak adlandırılır.
- **Popülasyon:** Doğal yaşamda popülasyonlar bireylerin bir arada bulunmasıyla oluşur. Olası çözüm bilgilerini içeren bireylerin bir araya gelmesiyle oluşan topluluk popülasyon olarak adlandırılır. Popülasyondaki birey sayısı çözüme kavuşturulacak problemin özelliğine göre, genetik algoritmayı tasarlayan uzman tarafından belirlenir.

2.1.1. Genetik Operatörler

Genetik operatörler, GA'nın temel yapısını oluşturan ve algoritmanın işleyişi sırasında mevcut popülasyon üzerinde uygulanan işlemlerdir. Bu operatörler, seçim (*selection*) ya da tekrar üreme (*reproduction*) operatörü, çaprazlama (*crossover*) operatörü ve mutasyon (*mutation*) operatörleridir (Elen & Çayıroğlu, 2010). Bunlara birlikte kısıtlı eniyileme problemlerinde kullanılması gereken ve probleme özgü olarak geliştirilen başka bir operatör de tamir (*reparation*) operatörüdür. Şimdi hiyerarşik düzene göre genetik operatörleri inceleyelim.

2.1.1.1 Seçim (Selection)

Seçim işlemi var olan bireyi genetik yapısında herhangi bir değişiklik yapmadan yeni nesile kopyalama işlemidir. Genel olarak “en iyi olan yaşar” prensibine dayanır. Burada amaç; oluşturulacak yeni nesilde daha yüksek uygunluk değerine sahip bireylerin sayısını arttırmaktır. Uygunluk değeri yüksek olan kromozomların sonraki nesile aktarılma olasılıkları daha yüksek olsa da seçme işlemi, daha düşük değerliğe sahip kromozomların da seçilmesine olanak sağlayacak şekilde dengeli olmalıdır. Aksi takdirde popülasyon tamamen iyi kromozomlardan oluşabilir. Bu durumda sonraki nesillerdeki kromozomların uygunluk değerlerinin iyileşmesini önleyecek farklılıklar kaybolabilir.

- **Rulet Tekeri:** Bütün kromozomlar uygunluk değerlerine göre bir rulet etrafında dizilirler. Rulet üzerinde uygunluk değerlerine göre sıralanan kromozomlar rasgele olarak seçilirler. Bu şekilde her birey seçilmek için kendi uygunluk değerine göre bu rulet tekerinden bir pay almaktadır. Daha büyük alana sahip bireyin seçilme şansı daha fazla olacaktır. Bu metot yardımıyla kromozomlar istatistiksel yöntemler kullanılarak uygunluk fonksiyonu değerlerinin toplam uygunluk fonksiyonuna oranları ölçüsünde seçilirler. Ancak bu seçim yönteminde uyum değeri büyük olan bireylerin seçilme olasılığı yüksek olduğu için, hep aynı kromozomların seçilmesine neden olmaktadır. Bu da popülasyon içindeki çeşitliliği etkileyerek sorun yaratır.
- **Sıralı (Rank):** Popülasyondaki kromozomlar uygunluk değerlerine göre büyükten küçüğe sıralanır. Kromozomun seçimansı, uygunluk değerinden çok oluşturulan liste içindeki yerine bağlıdır. En iyi kromozomdan başlanarak bir azalan işlev yardımıyla kromozomlara ait kopya sayısı belirlenir. Bir fonksiyon yardımıyla atanan kopya sayıları yeni jenerasyonun oluşturulmasında kullanılır.
- **Turnuva:** Bu seçim yönteminde, bireyler rastgele olarak gruplanır ve gruptaki bireyler aralarında seçim işlemi yapılmak üzere rekabete sokulur. Grup içinde en yüksek uygunluk değerine sahip olan birey, yeni nesli oluşturmak için ebeveyn bireylerden biri olarak seçilir. Bu işlem, toplam birey sayısına ulaşıncaya kadar devam eder. Bu yöntemde grup büyüklüğü önemlidir ve seçim yönteminin performansını önemli ölçüde etkiler. Bazı uygulamalarda grup büyüklüğü iki olarak seçilirken, bazılarında çok daha büyük gruplar oluşturulur. Turnuva seçim yöntemi, küçük popülasyonlu uygulamalarda, uygunluk değeri orantılı seçim yöntemlerinden daha iyi sonuç verir.
- **Sabit Durum:** Bu seçim operatöründe, ebeveynlerin seçimi için kromozomların büyük parçaları bir sonraki jenerasyona taşınmalıdır. Her nesilde yeni bir birey oluşturmak için büyük uygunlukta iyi olan birkaç kromozom seçilir. Daha sonra az uygunlukta bazı kromozomlar atılır ve yeni birey onların yerine getirilir. Popülasyonun geri kalanı değiştirilmeden yeni nesile aktarılır.

2.1.1.2 Çaprazlama (Crossover)

Çaprazlama, popülasyonda ebeveyn olarak seçilen iki kromozomun rastlantısal olarak belirli gen parçalarının karşılıklı yer değiştirilmesiyle yeni bireyler oluşturulması işlemidir. Çaprazlama operatörünün amacı, popülasyondaki kromozomların özelliklerini birleştirerek uygunluk değeri daha iyi olan bireyler elde etmektir ve temel parametresi çaprazlama olasılığıdır. Bu parametre çaprazlamanın hangi sıklıkta yapılacağını belirler. Çaprazlama olasılığı üreme havuzuna girecek kromozomların sayısını belirler. Bu oran yüksek olursa, iyi özellikteki bireylerin yeni popülasyonda

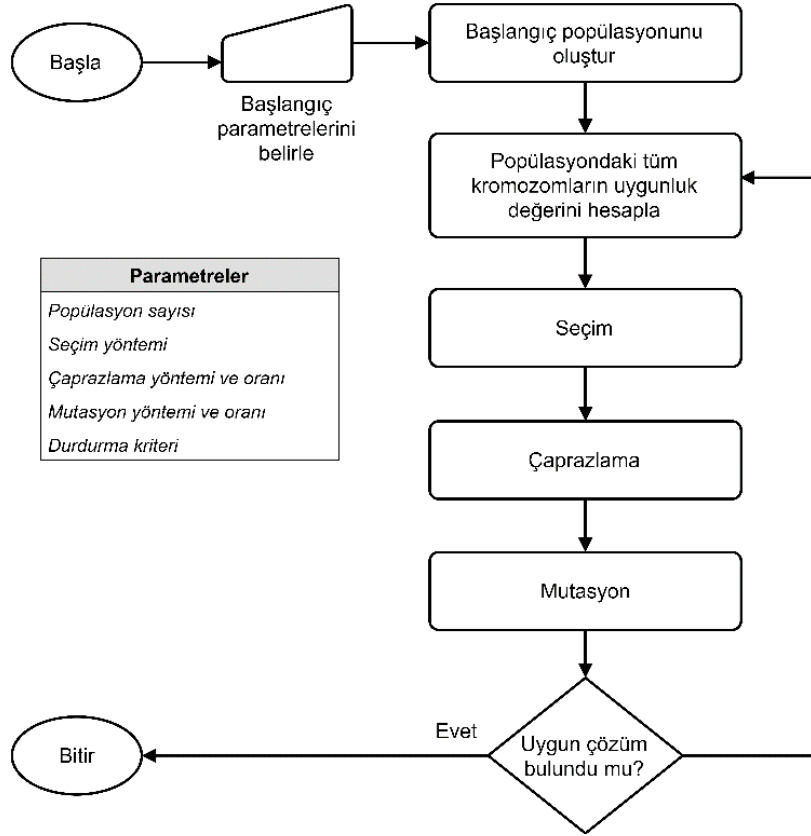
bulunma olasılığı azalır. Düşük olması da yeterli sayıda yeni bireyin oluşmasına mani olur (Haupt & Haupt, 2004). GA'da çaprazlama noktası ya da noktaları rastgele olarak belirlenir. Yaygın olarak kullanılan çaprazlama yöntemleri; tek noktali çaprazlama, iki noktali çaprazlama, tekdüze (*uniform*) çaprazlama, kısmen uyumlu çaprazlama (PMX), sıralı çaprazlama (OX), sıraya dayalı çaprazlama (OX2), dairesel çaprazlama (CX), konuma dayalı çaprazlama (POS), sezgisel çaprazlama ve maksimal koruyucu çaprazlama (MPX)'dir. Daha önceki çalışmamızda mevzu bahis çaprazlama yöntemleri detaylı olarak incelemiştir (Elen & Avuçlu, 2019).

2.1.1.3 Mutasyon (*Mutation*)

Popülasyonda çeşitliliğin sağlanması için mutasyondan faydalanılır. Mutasyon, mevcut bir bireyin genlerinin bir ya da birkaçının yerlerinin değiştirilmesi ile gerçekleştirilmektedir. Belirli bir iterasyon sonrasında popülasyon içerisindeki bireyler gitgide birbirlerine benzemektedir. Bu durum çözüm uzayının daralmasına neden olmaktadır. Bireylere ne kadar çaprazlama operatörü uygulansa da belli bir nesil sayısından sonra birey çeşitliliği sağlanamayabilmektedir. Bu durumda bireyi oluşturan genlerden rastgele bir tanesi seçilir ve bu genin değeri değiştirilir. Böylelikle popülasyon içindeki bireylerin çeşitliliğinin devamı sağlanmış olunur. Genellikle kullanılan mutasyon oranı, 1 sayısının kromozomdaki gen sayısına bölümü kadardır. Örneğin, 100 gen uzunluğuna sahip bir kromozom için bu oran 0.01'dir. Diğer bir deyişle, rastlantısal olarak düşünüldüğünde, her bir genin mutasyona uğrama olasılığı %1'dir (Back ve diğ., 2000).

2.1.2. GA'nın Çalışma Prensibi

GA için oluşturulan popülasyon çok sayıda bireyin bir araya gelmesiyle, yani çok sayıda olası çözüm adaylarının bir araya gelmesiyle oluşur (Çayıroğlu & Elen, 2012). Aday çözümler, probleme uygun şekilde kodlanmış diziler halinde tutulurlar. Bu diziyi oluşturan her bir eleman birey olarak adlandırılır. Her bir birey arama uzayında belirli bir bölgeyi temsil eder. Genetik algoritmada başlangıç bireyleri genelde rastgele üretilirler tabi bu işlem zorunlu değildir. Özellikle çok kısıtlı optimizasyon problemlerinde, başlangıç bireylerini oluşturmak için, tanımlanan kısıtlamaların bir kısmına dikkat edilerek daha iyi adaylar oluşturulabilir. Bireylerin, uygunluk fonksiyonu işlemine tabi tutulması sonucunda, çözümün optimal çözüme ne kadar yaklaştığını değerlendiren bir uygunluk değeri belirlenir. Başlangıç popülasyonu oluşturulmuş genetik algoritma üç evrim operatörüyle çalışmaktadır. Bunlar; seçim, çaprazlama ve mutasyon operatörleridir. Genel olarak bu operatörlerin her birisi, yeni nesilde oluşan popülasyonun her bireyine ayrı ayrı uygulanır. Seçim işlemi, popülasyondaki bireylerin uygunluk değerlerine bağlı olarak, yeni bireyler oluşturmak için, ebeveyn birey seçmesi işlemidir. Çaprazlama operatörü, seçim işleminin ardından uygulanır ve ebeveyn bireylere ait kromozomların belirli kısımlarının karşılıklı olarak yer değiştirmesini ve sonuçta yeni özellikte bireylerin oluşmasını ifade etmektedir. Mutasyon işlemi ise yeni oluşan bireyin kromozomlarından herhangi birinin içindeki bir geni mutasyon olasılığına bağlı olarak değiştirme işlemidir. Şekil 1'de GA'nın genel akış diyagramı gösterilmektedir.



Şekil 1: GA'nın akış diyagramı.

GA parametresinde tanımlanan popülasyondaki toplam birey sayısı kadar birey oluşturduğunda, yeni bir nesil meydana gelmiş olur. Bu yeni nesil önceki neslin yerini alır. Her nesilde bireylerin temsil ettiği çözüm değerlendirilerek, buldukları popülasyonun en iyi bireyi seçilir. Bulunan en iyi birey, önceki nesillerde bulunan en iyi bireylerden daha iyiyse onun yerine geçer. Genetik algoritma işlemini sonlandırmak için çeşitli yöntemler bulunmaktadır. Bu yöntemler; algoritmanın çalışması esnasında istenen çözüme ulaştığında, GA'nın başlangıcında tanımlanan toplam iterasyon sayısına ulaşıldığında veya uygunluk değeri sürekli olarak sabit kaldığında, bulunan en iyi bireyin temsil ettiği çözüm, problem için bulunmuş en uygun çözüm olarak sunulur.

2.2. Parçacık Sürüsü Optimizasyonu

Parçacık sürüsü optimizasyonu (PSO), 1995 yılında Kennedy ve Eberhart tarafından kuş sürüsü veya balık eğitiminin sosyal davranışlarından esinlenerek geliştirilen popülasyon tabanlı bir stokastik optimizasyon tekniğidir. PSO, önceden belirlenmiş bir kalite ölçüsüne göre aday çözümleri iteratif olarak geliştirmeye çalışarak problemi optimize eden bir hesaplama yöntemidir. Buradaki parçacıklar olarak adlandırılan aday çözümlerin her biri popülasyonun bir üyesidir. Popülasyondaki parçacıklar, konumu ve hızı üzerinde yapılan basit matematiksel formüllere göre arama uzayında hareket ettirerek problemi çözer. Her parçacığın hareketi, yerel olarak bilinen en iyi konumundan etkilenir. Ancak aynı zamanda, arama alanında diğer parçacıklar tarafından daha iyi konumlar bulunduğundan güncellenen en iyi bilinen konumlara doğru yönlendirilir. Bunun sürüyü en iyi çözümlere doğru hareket ettirmesi beklenmektedir.

PSO'nun, GA gibi evrimsel hesaplama teknikleriyle birçok ortak yönü bulunmaktadır. Bunlardan en önemlisi; tek bir bireysel uygulamaya odaklanmak yerine, bireylerin meydana getirdiği bir popülasyon (sürü) dikkate alınır. Sistem rastgele çözümlerden oluşan bir popülasyonla başlatılır ve nesiller güncellenerek optimal çözüm aranır. Bununla birlikte, GA'nın aksine PSO'nun geçit ve mutasyon gibi evrimsel operatörleri yoktur. PSO'da, parçacıklar adı verilen potansiyel çözümler, mevcut optimum parçacıkları takip ederek çözüm uzayında hareket ettirilir. GA ile karşılaştırıldığında, PSO'nun avantajları PSO'nun uygulanması kolay olması ve ayarlanması gereken birkaç parametre

olmasıdır. PSO, fonksiyon optimizasyonu, yapay sinir ağı eğitimi, bulanık sistem kontrolü ve GA'nın uygulanabileceği diğer alanlar gibi birçok alanda başarıyla uygulanmıştır. Tüm optimizasyon algoritmalarında olduğu gibi PSO'nun da kendine özgü terimleri vardır. Tablo 1'de PSO'da kullanılan terimler ve bunların açıklamaları verilmiştir.

Tablo 1: Parçacık Sürüsü Optimizasyonunda Kullanılan Terimler.

Popülasyon	Problemin çözümündeki bütün parçacıkların bir araya getirilmesiyle oluşturulan sürüye verilen isimdir.
Parçacık	Popülasyondaki (sürü) her bir bireyi temsil eder.
Hız (v)	Sürüdeki bir parçacığın, en iyi konuma ulaşmak için kullandığı bir vektördür.
Konum (x)	Mevcut problem için çözüm kümesini temsil eder.
Kişisel En İyi ($pbest$)	İterasyon süresince, herhangi bir parçacığın o ana kadar ulaştığı en iyi uygunluk değerine sahip konumunu temsil eder.
Global En İyi ($gbest$)	İterasyon süresince, sürünün o ana kadar ulaştığı en iyi uygunluk değerine sahip konumunu temsil eder. Bu konum, her bir iterasyonda sürüdeki parçacıkların $pbest$ değerine bakılarak tayin edilir. Yani, tüm zamanların en iyi uygunluk değerine sahip konum koşulu sağlandığı durumdur.
İterasyon	Belirli bir sayıda veya bir koşul sağlanıncaya kadar sürüdeki her bir parçacığa ait yapılması gereken işlemlerin tekrarlanmasıdır.
Uygunluk Fonksiyonu	Sürüdeki her bir parçacık konumunun uygunluk değerini hesaplamak için belirlenmiş bir kalite ölçüsüdür. Ayrıca literatürde amaç fonksiyonu olarak bilinir.

Sürü topolojisi, her bir parçacığın bilgi alışverişinde bulunabileceği parçacıkların alt kümesini tanımlar (Kennedy & Mendes, 2002). Bu topoloji, tüm parçacıkların diğer tüm parçacıklarla iletişim kurmasına izin verir ve böylece tüm sürü, tek bir parçacığın $gbest$ konumunu en uygun şekilde paylaşır. Bununla birlikte, bu yaklaşım sürünün lokal bir minimuma sıkışmasına neden olabilir (Mendes, 2004). Bu nedenle, parçacıklar arasındaki bilgi akışını kontrol etmek için farklı topolojiler kullanılmıştır. Örneğin, yerel topolojilerde parçacıklar sadece parçacıkların bir alt kümesiyle bilgi paylaşırlar. Yaygın olarak kullanılan bir sürü topolojisi, her bir parçacığın sadece iki komşusuna göre ilişkilendirilir, ancak bunun dışında diğer komşuluk durumları da vardır (Bratton & Kennedy, 2007). Bir topolojinin her zaman statik olmasına gerek yoktur. Aslında, topoloji parçacıkların iletişim çeşitliliği ile ilgili olduğundan (Oliveira ve diğ., 2016), uyarlanabilir topolojiler oluşturmak için bu konuda bazı çalışmalar (Almasi & Khooban, 2017; Miranda ve diğ., 2008; Clerc, 2006; Yin ve diğ., 2011; Elshamy ve diğ., 2007) yapılmıştır.

2.2.1. PSO'nun Çalışma Prensibi

Sistem rastgele potansiyel çözümlerden oluşan bir popülasyonla başlatılır. Literatürde yapılan bazı çalışmalar (Nguyen ve diğ., 2007; Pant ve diğ., 2008), başlangıç konumlarının uygun olmayan şekilde (düşük oranda düzenli bir dağılım kullanarak) oluşturulmasının etkili olabileceğini ileri sürdü. Bununla birlikte, diğer bazı çalışmalar, ilk dağılımın etkisinin birkaç yinelemeden (iterasyon) sonra çok hızlı bir şekilde azaldığını ve pratikte algoritmanın performansına önemli ölçüde bir etkisinin olmadığını göstermiştir (Omran ve diğ., 2013). Daha sonra, her potansiyel çözüm için rastgele bir "hız" atanır ve buna parçacık denir. Bu parçacıklar daha sonra potansiyel çözümlerin arama uzayında hareket ettirilir "uçurular". Bir sonraki aşamada, sürüde bulunan her bir parçacığın uygunluk değeri hesaplanır.

Her bir iterasyonda sürüdeki tüm parçacıkların uygunluk değeri önceki en iyi uygunluğu ($pbest$) ile karşılaştırılır. Geçerli değer $pbest$ 'ten daha iyiyse $pbest$ bu yeni değer ile güncellenir. PSO'da amaç sürüdeki en iyi konuma sahip parçacığının yerinin tespit edilip diğer parçacıkların da o yöne hareketinin sağlanmasıdır. Parçacıklar bir sonraki konumunu geçmiş tecrübelerine ve sürüdeki en iyi pozisyona sahip bireye dayanarak iyileştirmeyi hedefler. Benzer şekilde sürüdeki parçacıkların $pbest$ 'i birbirleriyle karşılaştırılır ve buna göre global en iyi uygunluk değeri ($gbest$) güncellenir. Daha sonra değişim hızı fonksiyonu ile her bir parçacığın yapacağı hareket belirlenir ve buna göre

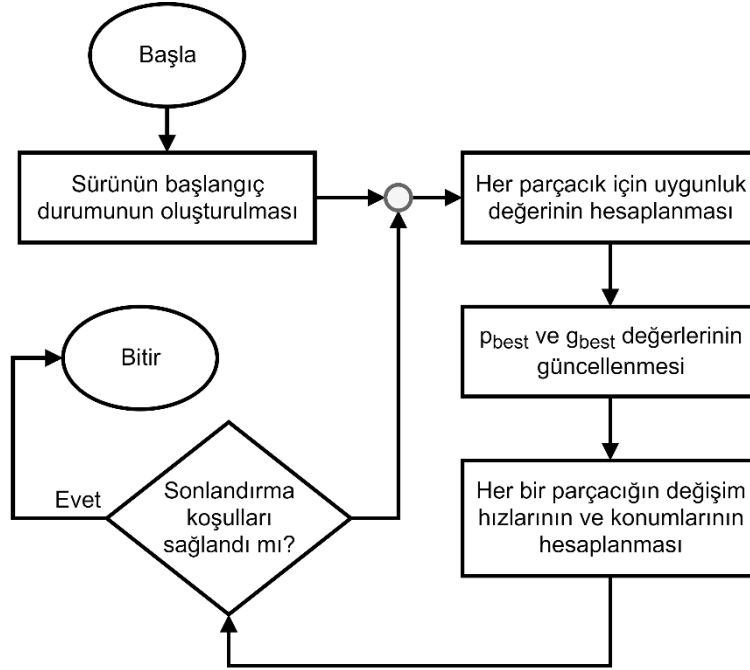
yeni durumları ayarlanır. Bu aşamadan sonra, tekrar uygunluk fonksiyonu ile sistemin çözüme ne kadar yaklaştığı denetlenir. Bu döngü arzu edilen koşullar yerine getirilinceye kadar tekrar edilir (Sharaf & Elgammal, 2018).

PSO'da bir parçacığın değişikliği yalnızca kendine ait hız vektörüne bağlıdır. Bu hız vektörü, parçacığın konumunu değiştirir ve yeni bir çözüm kümesi haline gelmesini sağlar. Parçacığın konumu (x), mevcut konumuna v hız vektörünün eklenmesiyle değiştirilir. Eşitlik 1.1'de i -inci iterasyondaki bir parçacığın hız vektörünün nasıl hesaplandığı gösterilmektedir. Buna ilaveten, Eşitlik 1.2'de parçacığın yeni konumu belirlenmektedir.

$$v_{n,i+1} = v_{n,i} + c_1 \times rand() \times (pbest_{n,i} - x_{n,i}) + c_2 \times rand() \times (gbest_i - x_{n,i}) \quad (1.1)$$

$$x_{n,i+1} = x_{n,i} + v_{n,i} \quad (1.2)$$

Burada v parçacık hızını, p mevcut parçacık değerini, c_1 ve c_2 öğrenme faktörlerini (genellikle $c_1 = c_2 = 2$ olarak belirlenir), $rand(0, 1)$ arasında üretilen rastgele bir sayıyı, $pbest$ bir iterasyondaki en iyi değere sahip çözümünü ve $gbest$ ise tüm zamanların en iyi değere sahip çözümünü temsil etmektedir. Prosedürün akış diyagramı Şekil 2'de gösterildiği gibidir

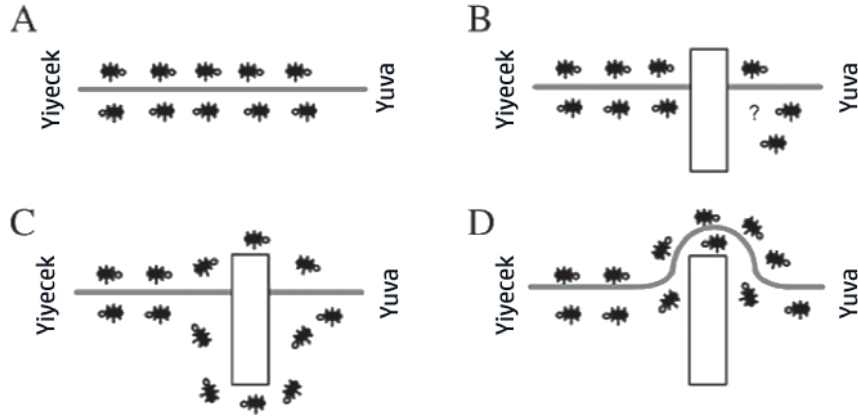


Şekil 2: PSO Algoritmasının Genel Akış Diyagramı.

2.3. Karınca Koloni Optimizasyonu

Sürü zekâsı temelli meta-sezgisel metotlardan biri olan Karınca Koloni Optimizasyonu (KKO), Dorigo tarafından 1991 yılında, günlük yaşamdaki karıncaların yiyecek arama davranışının modellenmesiyle geliştirilmiştir (Dorigo ve diğ., 1991). Model, gezgin satıcı problemi (GSP) ve kuadratik atama (KA) gibi NP-zor kapsamındaki kombinatoriyal optimizasyon çözümleri için tasarlanmıştır. Ancak KKO modelini kullanarak, sürekli optimizasyon problemlerine çözüm arayan çalışmalar da vardır. Karıncalar, buldukları besin kaynaklarını, en kısa yoldan yuvalarına taşımaya çalışırlar. Değişen çevre koşullarına hemen uyum sağlayabilirler. Dış etkenler neticesinde, karıncaların takip ettiği yol, artık en kısa yol değilse, besin kaynağı ile yuva arasında yeniden en kısa yolu bulabilirler. Şekil 3'te, gerçek karıncaların, besin kaynağı ile yuva arasında oluşturdukları yolun dış faktörlerden dolayı değişiminde, karıncaların, en kısa yolu yeniden nasıl bulduklarını anlatan kısa bir

tasarım görüntülenmektedir. Burada görüldüğü gibi, karıncalar, yuvalarına besin taşımak için düz bir hattı takip etmektedirler. Bu süreçte, sonradan gelen karıncaların da bu yolu bulabilmelerine yardımcı olması için feromon salgılamaktadırlar (Dorigo & Blum, 2005).



Şekil 3: Karıncaların en kısa yol bulma adımları.

Feromon, böceklerin, diğer böceklerle kokusal yolla, sosyal ilişki kurmak için yaydıkları, uçucu salgıya verilen genel isimdir. Ancak optimizasyon algoritmaları kapsamında, KKO algoritmalarıyla özdeşleşmiş ve bu algoritmaların en önemli bileşeni haline almıştır. Kullandıkları yol üzerine bir engel konulduğunda, karıncalar, gidecekleri yolu öncelikle rastgele seçmektedirler. Şekilden de görüldüğü üzere bir kısım karınca, engeli kuzeyden aşarken, bir kısmı güneyden aşmaktadır. Birim zamanda, kısa yolu kullanan karınca sayısı daha fazla olacağından, bu yolda biriken feromon konsantrisi daha yoğun olacaktır. Bu bağlamda, arkadan gelen karıncalar, tercihlerinde, feromon miktarını referans aldığından, en kısa yolu kullanan karınca sayısı zamanla artmış ve nihayetinde tüm karıncalar en kısa yolu kullanmaktadırlar. Başlangıçta rastgele yol seçimi yapan karıncaların, büyük olasılıkla feromon yoğunluğunu takip ederek, zamanla en kısa yolu bulması, kolektif bilginin ve bu bağlamda otokatalitik davranış modelinin en tipik örneklerindedir.

Meta-sezgisel yöntemlerin büyük bir çoğunluğu başlangıçta rastsal çözümler üretip, ardından kendi özel parametreleriyle bu başlangıç çözümlerinden daha başarılı çözümler türetmeye çalışırlar. Bu algoritmalar, başarılı bir keşif kabiliyetine sahiptirler. Yerel en iyi çözüme takılmadan, arama alanının farklı bölgelerine yayılabilmek için genellikle yeniden rastsal çözümler türetirler. KKO ise, çözüm bileşenleri arasındaki korelasyonu anlamaya çalışır. Bunun için feromon bilgisini kullanır. Çözüm bileşenleri arasındaki feromon miktarı, bu bileşenler arasındaki ilişki ile orantılıdır. Daha başarılı çözüm üretebilen parametreler arasındaki feromon konsantrisi yoğunlaştırılırken, başarısız çözüm oluşturan parametreler arasındaki feromon, iteratif olarak azaltılır. Diğer bir açıdan, KKO yaklaşımında feromon bilgisi, hafıza olarak kullanılır. KKO algoritmaları, feromon yaklaşımıyla güçlü bir sömürü yeteneği kazanmışlardır. Algoritma modeli, karınca sürülerinin yiyecek toplama davranışından ilham aldığı için, model karınca koloni optimizasyonu olarak adlandırılmıştır. Algoritma mantığında, karıncaları başarılı çözümler etrafında biriktirmek vardır. Dolayısıyla bu durum, algoritmanın keşif kabiliyetini önemli ölçüde etkiler. Bu sorunu aşmak için, algoritma farklı meta-sezgisellerle birlikte kullanılarak, birçok karma çözüm modeli de geliştirilmiştir. Feromon yoğunluğunu farklı formüllerle hesaplayan ya da geçiş kuralında, feromon değerini farklı şekillerde kullanan çok sayıda optimizasyon algoritması geliştirilmiştir. Ancak en belirgin ve en yaygın kullanım alanına sahip olanları karınca sistemi (KS), karınca koloni sistemi (KKS) ve Maksimum-Minimum Karınca Sistemi (MMKS)'dir. Dolayısıyla çalışma kapsamında bu algoritmalarından bahsedilecektir.

2.3.1. Karınca Sistemi (KS)

Karıncaların yiyecek arama davranışını modelleyerek geliştirilen ilk KKO algoritması karınca sistemi (KS)'dir. Algoritma, Dorigo ve arkadaşları tarafından GSP'yi çözmek için geliştirilmiştir (Dorigo ve diğ., 1991). Algoritmanın başlangıç aşamasında, m adet karınca düğümlere (V) rastsal olarak yerleştirilir ve düğümler arasına, 0.1 gibi eşit ve düşük değerlerde feromon $\tau(i,j)$ atanır.

Karınca, rastgele seçtikleri düğümlerle, turlarını tamamlayarak başlangıç düğümüne geri döner ve düğümler arasındaki (kenar) feromon miktarları güncellenir. Ardından iteratif olarak bu işlemler devam eder ve her bir çevrimde, o ana kadar bulunan genel en iyi çözüm hafızaya alınır.

Karınca, ziyaret edecekleri muhtemel düğümü seçerken, öncelikle, düğümlerin seçilme olasılıkları hesaplanır. Buna göre, i düğümünde bulunan bir karıncanın, bir sonraki düğüm için j düğümünü seçilme olasılığı Eşitlik 3.1’de gösterildiği şekilde belirlenir.

$$p_{i,j} = \begin{cases} \frac{[\tau(i,j)]^\alpha \times [\eta(i,j)]^\beta}{\sum_{u \in V} [\tau(i,u)]^\alpha \times [\eta(i,u)]^\beta}, & j, \text{ ziyaret edilmemiş bir düğüm ise} \\ 0, & \text{değilse} \end{cases} \quad (3.1)$$

Burada u , i -inci düğümünden gidilebilecek muhtemel düğümleri temsil eder. $i-j$ düğümleri arası mesafe $\delta(i,j)$ ’dir. $\eta(i,j)$ ise, bu mesafenin tersidir $\eta(i,j) = 1/\delta(i,j)$. Bir diğer parametre olan α feromon miktarını, β ise mesafenin göreceli önemini belirleyen sezgisel parametrelerdir.

Her bir karınca, turunu tamamladıktan sonra, düğümler arası feromon miktarları yeniden güncellenir. Feromon güncelleme işleminde öncelikle tüm düğümler arasındaki feromon konsantrisi, belirlenen oranda (ρ) buharlaştırılır. Ardından, her bir karıncanın geçtiği düğümler arasındaki feromon değeri, ilgili karıncanın kat ettiği mesafeye bağlı olarak artırılır. Bu sayede daha kısa yol kat eden karıncanın güzergahı üzerinde daha fazla feromon biriktirilmiş olur. Feromon güncelleme işlemlerinde Eşitlik 3.2’den yararlanır.

$$\tau(i,j) = (1-\rho) \times \tau(i,j) + \sum_{k=1}^m \Delta\tau(i,j) \quad (3.2)$$

Burada buharlaşma katsayısını temsil eden ρ parametresi, $[0,1]$ aralığında değerler alır. Diğer bir parametre olan $\Delta\tau(i,j)$ ise $i-j$ düğümleri arasındaki kenarı kullanan karıncanın, toplam yol güzergahıyla ilişkilidir. İlgili değer, Eşitlik 3.3’te gösterildiği şekilde hesaplanır.

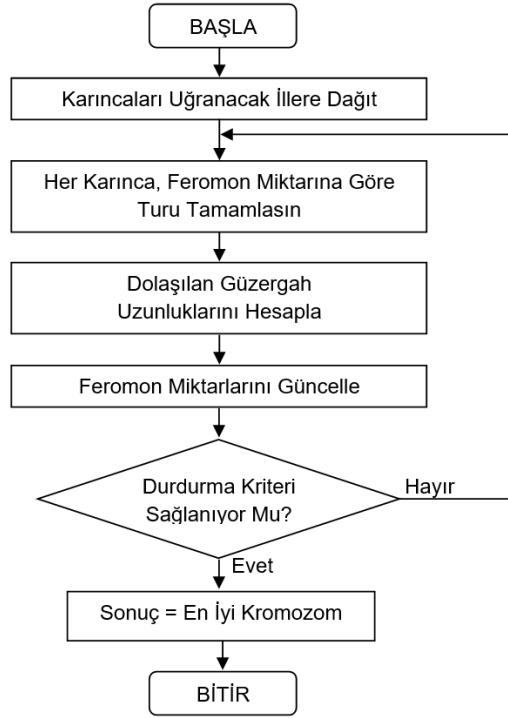
$$\Delta\tau(i,j) = \begin{cases} \frac{Q}{f(s_k)}, & k \text{ karıncası, } i-j \text{ kenarını kullanmış ise} \\ 0, & \text{değilse} \end{cases} \quad (3.3)$$

Burada Q , pozitif sabit bir sayı; $f(S_k)$ ise k karıncasının gittiği toplam yol mesafesi, diğer bir ifadeyle k -inci çözümün maliyetidir.

KS algoritması, KKO’nun en temel algoritmasıdır. Sonradan geliştirilen yöntemler KS temelinde, bu algoritmayı geliştirmek için yapılan güncellemeleri içerir.

2.3.2. Karınca Koloni Sistemi (KKS)

Karınca Koloni Sistemi (KKS), gerek feromon güncelleme ve gerekse geçiş kuralları açısından KS’den farklıdır. KKS algoritmasının akış diyagramı Şekil 4’teki gibidir.



Şekil 4: KKS algoritmasının akış diyagramı.

KKS, çözüm oluşturma sürecinde düğüm seçerken iki farklı alternatif kullanır. Alternatiflerin seçilme olasılığı $q_0[0,1]$ parametresiyle orantılıdır. Hangi alternatifin kullanılacağı ise, rastgele seçilen $q[0,1]$ değeriyle belirlenir. $q \leq q_0$ durumunda ilk alternatif uygulanır. q_0 için genellikle 1'e yakın bir değer atandığından, ilk alternatifin seçilme olasılığı yüksektir. Eşitlik 3.4 ile yapılan seçim, algoritmanın sömürü yeteneğini daha da güçlendirmektedir.

$$j = \max_{u \in V} \left\{ [\tau(i, j)]^\alpha \times [\eta(i, j)]^\beta \right\} \quad (3.4)$$

Eşitlik 3.4'te görüldüğü gibi sonraki düğüm, feromon matrisi, mesafe matrisi ve sezgisel parametreler dikkate alındığında, ilgili düğüm için en avantajlı düğümün seçimini gerektirir. Aksi durumda ($q > q_0$) ise, bir sonraki düğüm seçiminde Eşitlik 3.5'te gösterildiği gibi olasılık hesabı dikkate alınır.

$$p_{i,j} = \begin{cases} \frac{[\tau(i, j)] \times [\eta(i, j)]^\beta}{\sum_{u \in V} [\tau(i, u)] \times [\eta(i, u)]^\beta}, & j, \text{ ziyaret edilmemiş bir düğüm ise} \\ 0, & \text{değilse} \end{cases} \quad (3.5)$$

KKS kapsamında tüm karıncalar turlarını tamamladıktan sonra, düğümler arasındaki feromon miktarı iki şekilde güncellenir: yerel feromon güncelleme, genel feromon güncelleme. Yerel feromon güncelleme işlemi için Eşitlik 3.6 kullanılır.

$$\tau(i, j) = (1 - \rho) \times \tau(i, j) + \rho \times \tau(0) \quad (3.6)$$

Eşitlik 3.5'te KS algoritmasındaki feromon güncelleme işleminden farklı olarak, düğümler arasındaki feromon değeri buharlaştırıldıktan sonra, başlangıçta belirlenen feromon değeri $\tau(0)$, buharlaşma katsayısıyla çarpılarak artırılmaktadır. İşlemler, tüm karıncalar için tekrarlandıktan sonra, genel feromon güncelleme işlemi uygulanır.

Genel feromon güncelleme işlemi, yalnızca ilgili çevrime kadar en iyi çözüme ulaşan karınca için uygulanır. Bu sayede, en başarılı çözüme ait düğümler arasındaki feromon miktarları bir miktar daha artırılmış olur. Genel feromon güncelleme işlemi Eşitlik 3.7’de gösterildiği gibi hesaplanır.

$$\tau(i, j) = (1 - \alpha) \times \tau(i, j) + \alpha \times \Delta\tau(i, j) \quad (3.7)$$

Burada α , feromon eksiltme katsayısı; $f(S_{best})$, o çevrime kadar en başarılı karıncanın gittiği toplam yol mesafesi, diğer bir ifadeyle algoritmanın bulduğu optimal çözümünün maliyetidir. $\Delta\tau(i, j)$ değeri Eşitlik 3.8’deki gibi hesaplanır. Yerel feromon güncellemedekine benzer işlemler ilgili ana kadarki en başarılı karınca için tekrarlanır.

$$\Delta\tau(i, j) = \begin{cases} \frac{1}{f(S_{best})}, & \text{en iyi karınca, } i-j \text{ kenarını kullanmış ise} \\ 0, & \text{değilse} \end{cases} \quad (3.8)$$

2.3.3. Maksimum-Minimum Karınca Sistemi (MMKS)

Maksimum-Minimum Karınca Sistemi (MMKS) algoritması, KS’nin sömürü yeteneğini daha da güçlendirmek ve feromon değerlerini belli sınırlar içinde sınırlayabilmek için geliştirilmiş ve yaygın bir kullanım alanına sahip KKO algoritmasıdır. KS ve KKS algoritmalarında düğümler arasına yerleştirilen feromon miktarı, iteratif olarak her bir çevrimde değişir ve feromon değeri sınırlı değildir. MMKS, feromon değerlerini, problemin yapısına göre belli seviyelerde tutmayı amaçlar. Feromonun alt ve üst sınırlarını hesaplamak için farklı fomülasyonlar geliştirilmiştir. Bu formüllerden yaygın olarak kullanılanlardan birisinde, atanacak en büyük feromon değeri $\tau(t) = n / L_{min}$ eşitliğidir. Arama sürecinde t zamanda atanabilecek en büyük feromon değeri, toplam düğüm sayısının (n), t zamanına kadar bulunan en kısa tur mesafesine bölünmesiyle bulunur. Atanacak en düşük feromon değeri ise, $const / \phi \cdot n^2$ ile belirlenir. ϕ , ortalama tur mesafesidir.

MMKS algoritmasında her bir çevrimde, yalnızca en başarılı karıncanın feromon güncellemesine izin verilir. İlgili işlemler için Eşitlik 3.2 ve Eşitlik 3.3 kullanılır.

2.4. Yapay Arı Kolonisi Algoritması

Yapay Arı Kolonisi (YAK) algoritması 2005 yılında Derviş Karaboğa tarafından bal arılarının besin kaynağı arama kapsamındaki tavır ve etkileşimini referans alarak geliştirilen bir meta-sezgisel algoritmadır (Karaboğa, 2005). Algoritmada, besin kaynağı sayısı, işçi arı ve gözcü arı sayılarına eşittir. Algoritmada üç arı grubu bulunur bunlar: ziyaret edeceği besin kaynağı belli işçi arılar, yöneleceği besin kaynağını kendi tercihlerine göre belirleyen gözcü arılar ve rastgele kaynak arayışındaki kâşif arılar. Bal arıları, aralarındaki haberleşmeyi titreşim dans yaparak sağlar. Paylaşılan konum ve kalite bilgisi ile daha uygun yiyecek kaynakları bulunması hedeflenir (Grüter & Farina, 2009; Jiang, 2015). Kaynağın yerini öğrenen arı, güneş ışınları ile yörünge arasındaki açığı referans alarak ilgili kaynağa doğru hareket eder. Kaynaktan yiyecek toplamakla görevli arı nektar tükendiğinde kâşif arı olarak hareket etmektedir. Algoritma kapsamında, yiyecek kaynakların konumları problemin olası çözümlerini, nektar miktarı ise çözüm kalitesini temsil eder. Bal arıları en kaliteli kaynağı bulmayı amaçlamaktadır.

2.4.1. YAK’ın Çalışma Prensibi

Kâşif arılarca arama bölgesinde rastgele olarak besin kaynağı aranmasıyla başlayan algoritma, Eşitlik 4.1’de gösterildiği gibi yiyecek kaynaklarını rastgele yerleştirir.

$$s_{m,i} = l_i + rand(0,1) \times (l_i - u_i) \quad (4.1)$$

Burada S çözümler kümesi olmak üzere, $s_{m,i}$ m çözümünün, i -inci elemanına ait değerdir. l_i , ilgili çözüm bileşenine ait alt sınır, u_i ise üst sınırı ifade eder. İlk çözümler inşa edildikten sonra, çözümlerin, problemin uygunluk fonksiyonuyla hesaplanan göre, $f(s)$ değeri bulunur.

Başlangıçta oluşturulan bu kaynakların etrafında işçi arılarca bulunan diğer kaynak, Eşitlik 4.2 ile oluşturulur.

$$t_{m,i} = s_{m,i} + \phi_{m,i}(s_{m,i} - s_{r,i}) \quad (4.2)$$

Burada s_r rastsal seçilen herhangi bir çözümü temsil eder. i , ilgili çözümdeki rastsal eleman ve $\phi_{m,i}$ $[-1, 1]$ değer aralığındaki rastgele olarak belirlenen değerdir.

Oluşturulan yeni çözümün uygunluk değeri hesaplandıktan sonra sıradaki çözümünün uygunluk değeriyle karşılaştırılır. Tercih edilen çözüm, uygunluk değeri daha yüksek olandır. Eski çözümün uygunluk değeri daha yüksek ise, bu çözümün başarısızlık değeri +1 artırılır, aksi takdirde eski çözüm silinerek yerine oluşturulan yeni çözüm kaydedilir ve çözümün başarısızlık değeri sıfırlanır. s_m için uygunluk değeri $fit(s_m)$, uygunluk fonksiyonuna göre belirlenen $f(s_m)$ değerinden yararlanılarak, Eşitlik 4.3'te gösterildiği gibi hesaplanır.

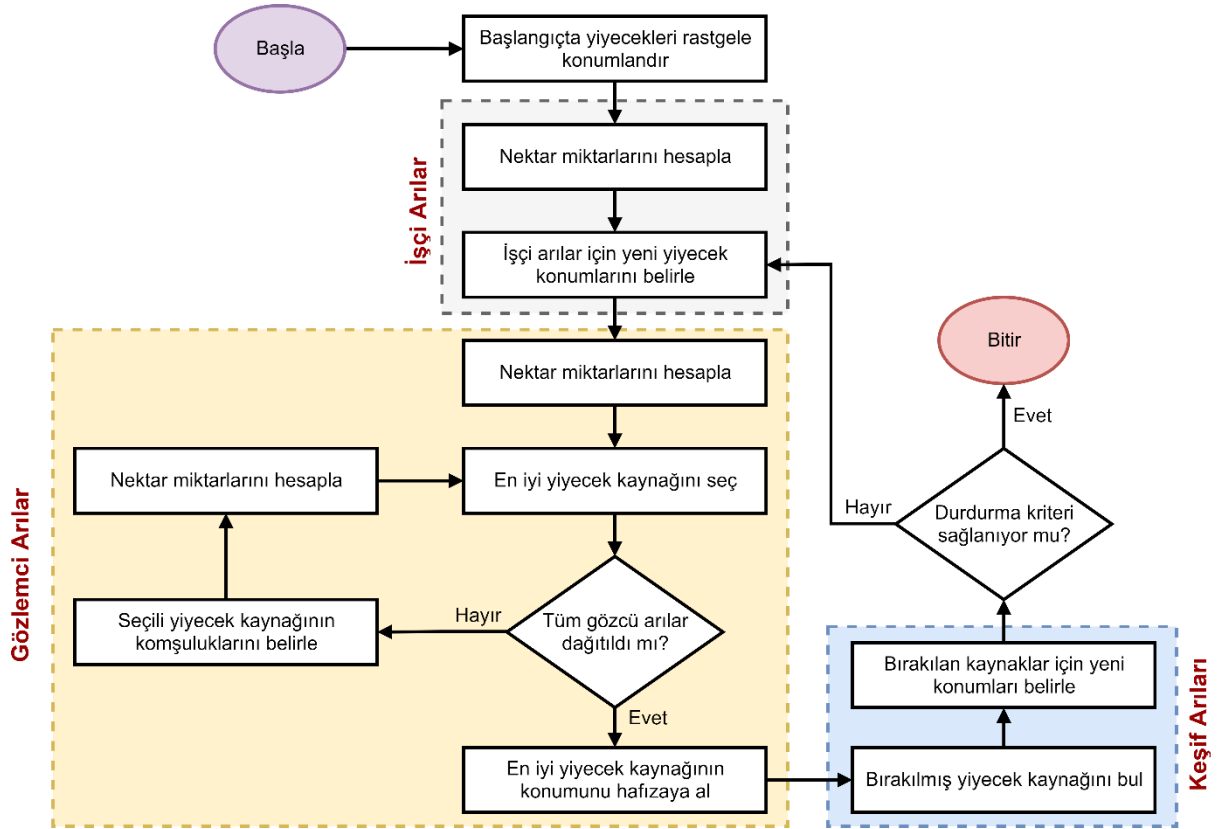
$$fit(s_m) = \begin{cases} \frac{1}{1 + f(s_m)}, & \text{eğer } f(s_m) \geq 0 \\ 1 + \text{abs}(f(s_m)), & \text{değilse} \end{cases} \quad (4.3)$$

Gözcü arıların çözüm arama yaklaşımı, işçi arılar gibidir. Ancak, işçi arılara kıyasla daha başarılı çözümler üretebilirler. Çünkü kaynak seçiminde, işçi arıların bilgilerinden yararlanırlar. İşçi arılar, gidecekleri besin kaynakları belli olan görevli arılardır. Gözcü arılar ise, işçi arıların bulunduğu daha başarılı çözümlere yönelme eğilimindedir. Standart YAK modelinde, gözcülerin işçi arıların bilgisinden yararlanması, rulet tekerleği ile sağlanır. Bu bağlamda işçi arıların tespit ettiği çözümler, uygunluk değerlerine göre rulet tekerleğine yerleştirilir. Gözcü arılar, çözüm türetmede ihtiyaç duydukları temel çözümleri rulet tekerleğinden seçer. Bu sayede, başarılı çözümlerin seçilme olasılığı artırılmış olur. YAK algoritması performansını geliştirmeye yönelik çalışmaların büyük bir bölümü, işçi ve gözcü arılar arasındaki bilgi paylaşımını güçlendirmeye yöneliktir.

İşçi arı safhasında tüm işçi arılarca çözümler oluşturulduktan sonra çözümlerin, gözcü arılar tarafından seçilme olasılığı hesaplanır. Her bir çözümün seçilme olasılığı uygunluk değerleriyle orantılıdır. Bu bağlamda, s_m çözümü için seçilme ihtimali Eşitlik 4.4'ten yararlanılır.

$$r_m = \frac{fit(s_m)}{\sum_{m=1}^{SN} fit(s_m)} \quad (4.4)$$

Gözcü arı aşamasında yeni bir çözüm türetilirken, işçi arı aşamasındakilere benzer işlemler yapılır. Farkı, çözüm üretirken yararlanılan her iki çözümün seçiminde Eşitlik 4.4'teki olasılık hesabının dikkate alınmasıdır. Seçilen çözümlere Eşitlik 4.2'de verilen denklem uygulanır ve yeni bir çözüm türetilir. Türetilen bu yeni çözümün uygunluk değeri Eşitlik 4.3 ile belirlenir ve yine aç gözlü yaklaşım uygulanır. Kâşif arı aşamasında, başarısızlık değeri limit' seviyesine gelen çözümler silinir ve bunların yerine Eşitlik 4.1'deki eşitlik kullanılarak rastsal çözümler türetilir. Şekil 5'te algoritmanın akış diyagramı gösterilmektedir.



Şekil 5: YAK'ın akış diyagramı.

3. SONUÇLAR

Son 30 yılda karmaşık optimizasyon problemlerini çözmek için çok sayıda meta-sezgisel yaklaşım geliştirildi. Başarıları, büyük ölçüde en önemli özelliklerinden, yani optimizasyon problemi hakkında minimum ek bilgiye ihtiyaç duyulması ve algoritmaların son derece sayısal sağlamlığından kaynaklanmaktadır (Gavrilas, 2010). Bu çalışma, en popüler meta-sezgisel optimizasyon tekniklerine ve bunların optimizasyon problemlerine nasıl uygulandığına ilişkin temel bilgileri sağlamıştır. Buna ek olarak, optimizasyon problem çözümleri alanındaki araştırmacılara, algoritma tercihinde ışık tutan bir analiz hedeflenmiştir. Kapsam olarak, çözüm arama sürecinde farklı stratejiler kullanan, yaygın olarak tercih edilen ve uygulandıkları optimizasyon problemlerinde başarılı çözümler üretebilen, popülasyon temelli meta-sezgisel algoritmalar incelenmiştir. Algoritmik yapı itibarıyla kombinatoriyal ve sayısal optimizasyon problem çözümlerine yönelik geliştirilen algoritmalar, ayrı ayrı ele alınmıştır. Farklı bir açıdan; global optimumu popülasyon içinde tutan GA ve PSO, genel optimumu hafızasında tutan ve yerel optimum aralığına sıkıştığında yeniden rastsal çözümler oluşturan YAK ve çözümleri bütünler halinde incelemek yerine, çözüm bileşenleri arasındaki korelasyonu inceleyerek optimal çözümü inşa etmeye çalışan KKO metodlarına değinilmiştir. Algoritmaların kullandığı formülasyonlar ya da nesne dizilişini değiştirme yaklaşımları paylaşılmış, anlatım akış diyagramları ile detaylandırılmıştır.

4. KAYNAKÇA

Almasi, O. N. & Khooban, M. H. (2017). A parsimonious SVM model selection criterion for classification of real-world data sets via an adaptive population-based algorithm. *Neural Computing and Applications*, 1-9. doi: 10.1007/s00521-017-2930-y

Altunbey F. & Alataş B. (2015). Sosyal Ağ Analizi için Sosyal Tabanlı Yapay Zekâ Optimizasyon, *IJPAS*, 1(1), 33-52.

Back, T., Fogel, D. B. & Michalewicz, T. (2000). *Evolutionary Computation 1: Basic Algorithms and Operators*, Institute of Physics Publishing, Bristol and Philadelphia.

Blum, C. & Rolli, A. (2003). Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison, *ACM Computing Surveys*, 35(3), 268–308.

Bratton, D. & Kennedy, J. (2007). Defining a Standard for Particle Swarm Optimization, *Proceedings of the 2007 IEEE Swarm Intelligence Symposium (SIS 2007)*. pp. 120–127. doi: 10.1109/SIS.2007.368035

Çayıroğlu, İ. & Elen, A. (2012). A Heuristic Optimization Approach for A Real-World University Timetabling Problem, *Advances in Computer Science and Engineering*, vol. 9(2), pp. 103-131.

Clerc, M. (2006). Particle Swarm Optimization. ISTE (International Scientific and Technical Encyclopedia).

Dorigo, M., Maniezzo, V. & Colorni, A. (1991). Positive Feedback as a Search Strategy, Technical Report n. 91-016, Politecnico di Milano.

Dorigo, M. & Blum, C. (2005). Ant colony optimization theory: A survey, *Theor. Comput. Sci.*, 344(2–3), pp. 243–278.

Elen, A. & Avuçlu, E. (2019). Genetik Algoritmada Çaprazlama Yöntemleri, *Geleceğin Dünyasında Bilimsel ve Mesleki Çalışmalar 2019: Bilgisayar Mühendisliği*, Ekin Basım Yayın Dağıtım, 1-16.

Elen, A. & Çayıroğlu, İ. (2010). Solving of Scheduling Problem with Heuristic Optimization Approach, *Journal of Technology*, vol. 13(3), pp. 159-172.

Elshamy, W., Rashad, H. & Bahgat, A. (2007). Clubs-based Particle Swarm Optimization, *IEEE Swarm Intelligence Symposium 2007 (SIS2007)*. Honolulu, HI. pp. 289–296.

Gavrilas, M. (2010). Heuristic and metaheuristic optimization techniques with application to power systems. In *Proceedings of the 12th WSEAS international conference on Mathematical methods and computational techniques in electrical engineering (MMACTEE'10)*. Wisconsin, USA, 95–103.

Glover, F. (1986). Future Paths for Integer Programming and Links to Artificial Intelligence, *Computers and Operations Research* 13 (5): 533–549.

Goldberg, D. E. (1989). *Genetic Algorithms in Search Optimization and Machine Learning*, Addison Wesley Publishing Company, U.S.A., 1-10.

Grüter, C. & Farina, W. M. (2009). The honeybee waggle dance: can we follow the steps? *Trends in Ecology & Evolution*, 24(5), 242–247. doi: 10.1016/j.tree.2008.12.007

Haupt, R.L. & Haupt, S.E. (2004). *Practical genetic algorithms*, Second Edition, New Jersey: Wiley-Interscience.

Jiang, H. (2015). Artificial Bee Colony algorithm for Traveling Salesman Problem.

Karaboğa, D. (2005). An Idea Based on Honeybee Swarm for Numerical Optimization, Technical Report TR06, Erciyes University, Engineering Faculty, Computer Engineering Department.

Kennedy, J. & Eberhart, R. (1995). Particle Swarm Optimization. *Proceedings of ICNN'95-International Conference on Neural Networks*, Perth, WA, Australia, 4, 1942-1948. doi: 10.1109/ICNN.1995.488968

Kennedy, J. & Mendes, R. (2002). Population structure and particle swarm performance, *Evolutionary Computation, CEC'02. Proceedings of the 2002 Congress on*. 2. pp. 1671–1676 vol.2. doi: 10.1109/CEC.2002.1004493

Li, Y., Yang, Y., Zhou, L. & Zhu, R. (2009). Observations on Using Problem-Specific Genetic Algorithm for Multiprocessor Real-time Task Scheduling, *International Journal of Innovative Computing, Information and Control*, 5(9): 1349-4198.

Mendes, R. (2004). *Population Topologies and Their Influence in Particle Swarm Performance* (PhD thesis), Universidade do Minho.

Miranda, V., Keko, H. & Duque, Á. J. (2008). Stochastic Star Communication Topology in Evolutionary Particle Swarms (EPSO), *International Journal of Computational Intelligence Research (IJCIR)*, Volume 4, Number 2, pp. 105-116.

Nguyen, X.H., Nguyen, Q.U. & McKay, R.I. (2007). PSO with randomized low-discrepancy sequences. In: *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation, GECCO '07*, pp. 173–173. ACM, New York. doi: 10.1145/1276958.1276987

Oliveira, M., Pinheiro, D., Andrade, B., Bastos-Filho, C.& Menezes, R. (2016). Communication Diversity in Particle Swarm Optimizers, *International Conference on Swarm Intelligence. Lecture Notes in Computer Science, 9882*. pp. 77–88.

Omran, M.G.H., al Sharhan, S., Salman, A. & Clerc, M. (2013). Studying the effect of using low-discrepancy sequences to initialize population-based optimization algorithms. *Computational Optimization and Applications, 56(2)*, 457–480. doi: 10.1007/s10589-013-9559-2

Pant, M., Thangaraj, R., Grosan, C. & Abraham, A. (2008). Improved particle swarm optimization with low-discrepancy sequences. *IEEE Congress on Evolutionary Computation (CEC2008)*, pp. 3011–3018. doi: 10.1109/CEC.2008.4631204

Sharaf, A. M., & Elgammal, A. A. A. (2018). Novel AI-Based Soft Computing Applications in Motor Drives. *Power Electronics Handbook*, 1261–1302. doi:10.1016/b978-0-12-811407-0.00042-8

Talbi, E-G. (2009). Population-Based Metaheuristics. *Metaheuristics: From Design to Implementation*, John Wiley & Sons, 190–307. doi: 10.1002/9780470496916.ch3

Yin, P., Glover, F., Laguna, M., & Zhu, J. (2011). A Complementary Cyber Swarm Algorithm. *International Journal of Swarm Intelligence Research (IJSIR)*, 2(2), 22-41.

FEROMONAL YAPAY ARI KOLONİ ALGORİTMASI (fYAK)

Dursun Ekmekci*, Abdullah Elen

**Karabük Üniversitesi T.O.B.B. Teknik Bilimler M.Y.O. Bilgisayar Teknolojileri Bölümü, Karabük*

dekmekci@karabuk.edu.tr

** Sorumlu Yazar*

1. GİRİŞ

“Sürü” terimi, kolektif davranış sergileyen hayvan ve böcek kümeleri için kullanılır. Koloni halinde yaşayan bu canlılar için iki temel kavram vardır: kendi başına organize olma ve kolonideki diğer canlılarla iş bölümü. Kendi başına organize olabilme; sistemdeki diğer birimlerle etkileşimden alınan bilgileri kullanarak, kendi başına sistemin bütününe etkilemektir (Karaboğa & Akay, 2007). İş bölümü ise, topluluktaki bireylerin eş zamanlı olarak farklı işleri gerçekleştiriyor olmasıdır. Bu iş bölümünün, görevlerin uzman olmayan kişiler tarafından sırayla gerçekleştirilmesinden daha verimli olduğuna inanılmaktadır (Barnebau ve diğ., 1999). Koloni bireylerinin kendi aralarında ve çevreyle etkileşimi, diğer yandan kolektif zekanın oluşumuna vesile olur. Teknik alanda "sürü zekâsı" ifadesi ilk olarak Beni ve Wang tarafından hücrel robotik sistem için kullanılmıştır (Beni & Wang, 1993). Hücrel robotik sistemlerinde, birçok basit ajan, desenler oluşturmak için bir veya iki boyutlu ortamları işgal etmiş ve en yakın komşu etkileşimleriyle kendi başlarına organize olmuşlardır. Daha sonra sürü zekâsı, bilgisayar bilimcileri, ekonomistler, bioinformatikçiler, mühendisler, yöneylemciler ve diğer birçok disiplin için giderek daha önemli bir araştırma alanı haline geldi (Karaboğa ve diğ., 2014). Yapay zekâ araştırmacıları, aralarında iş bölümü olan canlıların koloni halindeki yaşamını modelleyerek sürü zekâsı temelli optimizasyon algoritmaları geliştirdi. Bu bağlamda kuş ve balık sürülerinin (Kennedy & Eberhart, 1995), karınca kolonilerinin (Dorigo ve diğ., 1991), ateş böceklerinin (Yang, 2009), guguk kuşlarının (Yang & Deb, 2009) sosyal hayatlarını matematiksel olarak modelleyerek optimizasyon algoritmaları geliştirilmiştir.

Koloni yaşamının diğer bir popüler örneği de bal arısı sürüleridir. Bir bal arısı kolonisi, çevreden bilgi toplayan ve davranışını buna göre düzenleyen dinamik bir sistem olarak kabul edilir (Tereshko, 2000). Aralarında herhangi bir hiyerarşi olmadan, kolektif davranışa dayalı, organize olabilme yeteneği ve bu bağlamda kurulan iş bölümü, sürü zekâsının en temel niteliklerindedir. Bu yaşam biçimi, bal arılarını, farklı yön ve alanlardaki birçok araştırmacı için önemli kılmıştır. Örneğin Avusturyalı bilim insanı Karl V. Frisch, bal arılarının birbirleriyle haberleşmede titreşim yöntemini kullandıklarını keşfetmiş ve 1973 yılında Nobel Fizyoloji veya Tıp ödülüne layık görülmüştür. Biyolojiden sağlığa, sosyolojiden kimyaya, hemen her alanda üzerinde çalışmalar yürütülen bu mikroorganizmalar, mühendislik disiplinlerince de dikkatle incelenmektedir. Bal arılarının yem arama, etkileşim, iletişim ve evlilik ilişkileri gibi sosyal davranışlarına dayanan; kombinatoriyal optimizasyon, sayısal fonksiyon optimizasyonu, bilgisayar ağı ve mobil ağlar için algoritmalar geliştirilmiştir (Abbass, 2001; Pham ve diğ., 2005; Yang, 2005). Bal arılarının yiyecek arama davranışını modelleyen sürü zekâsı temelli bir optimizasyon algoritması da Karaboğa tarafından 2005 yılında literatüre tanıtılan Yapay Arı Koloni (YAK) algoritmasıdır (Karaboğa, 2005). Gerçek hayatta bal arıları, bal yapımı sürecinde şu faaliyetleri yürütür: rastgele yiyecek arama, bulunan yeni besin kaynakları hakkındaki bilgileri, kovadaki diğer arılarla paylaşma, daha kaliteli besin kaynağı bulma çabası, nektarı tükenmiş besin kaynağını terk etme ve yeniden yiyecek arama. Tereshko ve Loengarov, bu süreci, reaksiyon yayılma denklemlerine dayanan bu model olarak özetlemişlerdir (Tereshko & Loengarov, 2005). YAK, yiyecek toplama serüvenindeki bu faaliyetlerin tamamını modellemeye çalışır.

Çalışma kapsamında, YAK algoritmasının yerel arama ve yakınsama performansını artırmak için geliştirilen Feromonal Yapay Arı Koloni (fYAK) algoritması açıklanmaktadır. Bu kapsamda, öncelikle YAK algoritması detaylı olarak ele alınmış, algoritma adımlarındaki işlemler aktarılmıştır.

Ardından, algoritmanın feromonal modeline ilham olan yaklaşım, Karınca Koloni Sistemi (KKS) algoritması üzerinden açıklanmıştır. Sonraki bölümde, FYAK tanıtılmış, ayrık yapılı ve sayısal optimizasyon problemleri için geliştirilen tasarımları paylaşılmış, test problemlerinden elde edilen sonuçlar sunulmuştur.

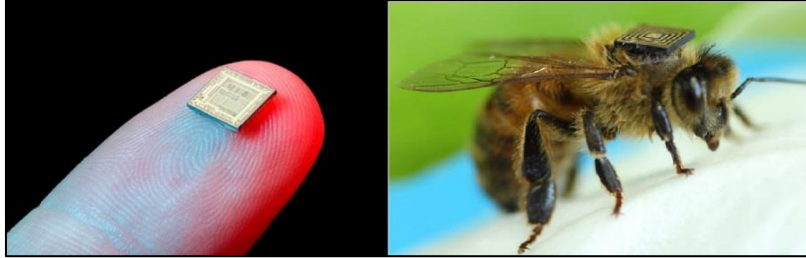
2. YAPAY ARI KOLONİ ALGORİTMASI

YAK algoritması, popülasyona dayalı, sürü zekâsı temelli bir algoritmadır. Algoritmanın standart tasarımı, sayısal optimizasyon problemlerine yönelik hazırlansa da ayrık yapılı optimizasyon problem çözümleri için geliştirilmiş tasarımlar da mevcuttur (Karaboğa ve diğ., 2014). Kolay kodlanabilen, esnek yapısıyla algoritma, farklı türden optimizasyon problemlerine rahatlıkla uygulanabilir. Diğer bir avantajı ise yalnızca iki tane kontrol parametresi kullanıyor olmasıdır.

Bu bölümde, öncelikle gerçek yaşamdaki bal arılarının, yiyecek arama sürecinde YAK algoritmasına ilham kaynağı olan iş birliğine değinilecek, ardından, algoritma modeli, adımlarıyla birlikte açıklanacaktır.

2.1. Bal Arısı Kolonisinin Yiyecek Toplama Sürecindeki Kolektif Davranışı

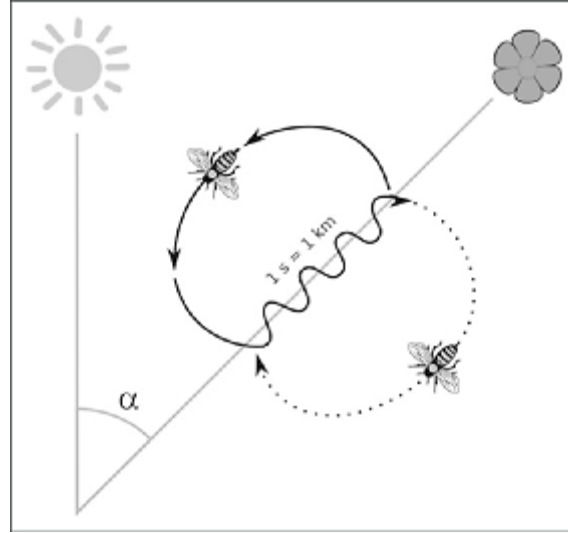
Bal arıları, sosyal aktivitelerinin hemen hepsinde iş bölümüyle hareket eden canlılardır. Her kovan, farklı bir koloniyi temsil eder. Bir kovanda, yalnızca bir tane kraliçe arı bulunmasına rağmen, kolonide toplam 5000 ile 80000 arasında bal arısı yaşar. Erkek arılar üremeden sorumludur, yiyecek toplama sürecinde görevli arılar dişi arılardır. Araştırmalar, toplam koloni popülasyonunun, %5'i ile %10'unun yeni besin kaynağı bulmakla görevli olduğunu göstermektedir. Araştırmacılar, bal arılarının, yiyecek arama faaliyetindeki yönelim ve tercihlerini analiz etmek üzere, Şekil 1'de görüldüğü gibi, bal arılarının sırt bölümlerindeki tüylü yüzeye mikroçip yerleştirmiş ve kovandan gönderdikleri RF sinyalleriyle, uçuş hareketlerini takip etmişlerdir. Deneyler sonucunda, yeni bir besin kaynağı keşfetmek üzere kovandan ayrılan bir bal arısının, kovandan 15 km kadar uzağa gidebildiği ve bu seyahatinde 150 kadar besin kaynağının uygunluğunu kontrol edebildiği gözlemlenmiştir.



Şekil 1: Bal arısının, yiyecek toplama sürecindeki hareketlerini takip için yapılan çalışmalar

Polen toplamak üzere uygun bir besin kaynağı keşfedildikten sonra, arı, ilgili besin kaynağına ait bilgileri, kovandaki diğer arılarla da paylaşır. Bu iletişim için bal arıları, titreşim yöntemini kullanırlar. Kovana dönen arı, önce keşfettiği kaynaktan getirdiği besinin bir parçasını, kovandaki arılara tattırır ve ardından titreşimler yaparak 8 şekline benzer figürler çizer. Arı, 15 saniyelik periyotlarda ne kadar çok titreşim yaparsa, mesaj, “bulunan kaynak, kovana daha yakın mesafededir” anlamına gelir.

Titreşim hareketinin yönü ise, keşfedilen besin kaynağının yönünü ifade eder. Yeni besin kaynağı bulan arıyı takip eden gözcü arılar, güneş ışınlarının kovana dik geldiği doğruyu referans alır ve titreşim yönü kadar açıyla yönelerek uçuşa geçerler. İlgili iletişim yöntemi Şekil 2’de şematize edilmiştir.



Şekil 2: Bal arılarının titreşim yaparak iletişim besin kaynağı tarifi

Bulunan besin kaynağının kovana mesafesi kadar, çokluğu, kalitesi, toplama kolaylığı gibi unsurlar da titreşim tarzını belirleyen diğer parametrelerdir. Araştırmacılar, bal arılarının titreşim hareketinden farklı olarak, ses çıkarma ve feromon salgılama gibi yöntemlerle de iletişim kurduklarını gözlemlemişlerdir.

2.2. YAK Algoritması Ana Modülleri

Gerçek bal arısı kolonilerinin yiyecek arama davranışını modelleyen YAK algoritması kapsamında temel öğeler, besin kaynakları ve bal arıları olarak özetlenebilir. Besin kaynakları çözüm alanındaki muhtemel çözümleri temsil ederken, bal arıları en kaliteli besin kaynağını arayan prosedürlerdir. Yiyecek arama sürecinde, rastgele besin kaynakları arayan *kâşif arılar*, keşfedilen besin kaynaklarından besin toplamakla görevli *işçi arılar* ve işçi arıların tecrübelerinden yararlanarak daha kaliteli besin kaynaklarına yönelmeye çalışan *gözcü arılar* bulunur. Kullanıcının seçebileceği değerler, *besin kaynağı* ve *limit* parametre değerleridir. Algoritma kabulünde işçi ve gözcü arı sayıları, besin kaynağı sayısına eşittir. Bu bağlamda koloni boyutu, besin kaynağı sayısının iki katı kadardır. Kâşif arılar, koloninin üçüncü bir grubu değil, işçi ve gözcü arıların, rastgele kaynak aramak için üstlendikleri görevlidir.

YAK algoritması en temelde şu safhalardan oluşur (Karaboğa, & Basturk, 2007):

- Kâşif arı safhası

Tekrarla

- İşçi arı safhası
- Gözcü arı safhası
- En kaliteli besin kaynağının hafızaya alınması
- Limit kontrolü ve kâşif arı oluşumu

Ta ki şartlar sağlanıncaya kadar

2.2.1. Kâşif Arı Safhası

Algoritmanın başlangıcında, kâşif arılarca, belirlenen çözüm sayısı (ζS) kadar, rastgele çözümler oluşturulur. ζ_a , çözüm matrisinin a -ıncı satırındaki çözüm, ES çözümdeki eleman sayısı, b çözümün sıradaki elemanı ($1..ES$) ve $\zeta_{min b} - \zeta_{max b}$ b 'nin alt ve üst sınırları olmak üzere, çözüm matrisi Eşitlik 1'de gösterildiği gibi tamamlanır.

$$\zeta_{a,b} = \zeta_{min b} + rast(0,1) * (\zeta_{max b} - \zeta_{min b}) \quad 1)$$

Her bir başlangıç çözümü oluşturulduktan sonra, ilgili çözümün, amaç fonksiyonuna göre amaç değeri ($f(\zeta_a)$) hesaplanır ve ardından, uygunluk değeri belirlenir. Çözümlerin uygunluk değeri hesaplanırken Eşitlik 2'den yararlanır.

$$fit(\zeta_a) = \begin{cases} 1/(1 + f(\zeta_a)) & \text{Eğer } (f(\zeta_a)) \geq 0 \\ 1 + abs(f(\zeta_a)) & \text{Eğer } (f(\zeta_a)) < 0 \end{cases} \quad (2)$$

2.2.2. İşçi Arı Safhası

İşçi arı safhasında, sırasıyla her bir başlangıç çözümünün komşuluğunda yeni çözümler aranır. Yeni çözüm oluşturulurken Eşitlik 3'ten yararlanır.

$$T_{a,b} = \zeta_{a,b} + \phi_{a,b}(\zeta_{a,b} - \zeta_{k,b}) \quad (3)$$

Eşitlik 3'te ζ_a sıradaki çözüm olmak üzere, ilgili çözümün b . elemanına, rastgele seçilen ζ_k çözümünün b . elemanı arasındaki fark, $[-1,1]$ aralığında rastgele seçilen ϕ değeri ile çarpılarak eklenir.

Yeni çözüm oluşturulduktan sonra, türetilen bu çözüm (T_a) ile temel çözüm (ζ_a) arasında aç gözlü yaklaşım uygulanır. Bu bağlamda, eğer yeni çözüm mevcut çözümden daha başarılı değilse, ilgili çözüm, daha başarılı bir çözüm türetemediğinden, başarısızlık sayacı 1 artırılır. Aksi takdirde, mevcut çözüm yerine yeni çözüm kaydedilecek ve ilgili çözümün başarısızlık değeri sıfırlanacaktır. Karşılaştırmada kullanılan başarı kriteri, çözümlerin uygunluk değerleridir.

2.2.3. Gözcü Arı Safhası

Bahsedildiği üzere YAK mantığında, gözcü arılar, kaynak seçiminde işçi arıların bilgisinden yararlanır ve yönelecekleri besin kaynaklarını kendileri seçerler. Algoritmik boyutta bilgi aktarımı, rulet tekerine dayanan tecrübe paylaşımına dayanır. İşçi arıların kovana dönmesinden sonra, elde edilen güncel çözümler rulet tekerleğine yerleştirilir. Gözcü arılar, yeni çözümler türetirken, rulet tekerinden seçtikleri çözümleri kullanırlar. Bu sayede, arama daha başarılı çözümler etrafında yoğunlaşır.

Gözcü arılar için, her bir çözümün seçilme olasılığı (r_a) Eşitlik 4 ile hesaplanır.

$$r_a = \frac{fit(\zeta_a)}{\sum_{a=1}^{CS} fit(\zeta_a)} \quad (4)$$

Gözcü arı safhasında da yeni çözümler türetilirken Eşitlik 3'ten yararlanır. Yine türetilen çözümle, mevcut çözüm arasında aç gözlü yaklaşım uygulanarak, algoritmanın daha başarılı çözümlere yönelmesi hedeflenir.

İşçi arılarla gözcü arıların haberleşmesinde birçok yöntem geliştirilmiştir. İşçi arı safhasında, türetilen çözümle mevcut çözüm arasındaki farklı inceleyen, gözcü arı safhasındaki arama yönünü işçi arı safhasında elde edilen çözümlere göre belirleyen, daha çabuk çözüm üreten yöntemler örnek olarak verilebilir. Bu noktada amaç, işçilerin bilgisinden daha çok yararlanabilmektir. Çalışma kapsamında anlatılan Feromonal Yapay Arı Koloni (fYAK) algoritması da bu amaçla geliştirilen yöntemlerden biridir. fYAK, gözcü arı safhasında rassal çözüm bileşenleriyle yeni çözümler türetmek yerine, işçi arı safhasında elde edilen çözüm bileşenlerinin en uygun kombinasyonunu oluşturmaya çalışır. Bir sonraki bölümde, yaklaşım modeli ve algoritmik tasarımı detaylı olarak açıklanacaktır.

2.2.4. Başarısızlık Kontrolü ve Rastgele Çözümler Oluşturma

Kâşif arılar, algoritma, belli çevrim sayısınca daha başarılı çözümler türetmediğinde, yerel en iyi çözüme takılmamak için, aramayı farklı alanlara yönlendirmeye çalışan prosedürlerdir. İşçi arı ve gözcü arı safhalarındaki çözüm türetme işlemleri tamamlandıktan sonra, çözümlerin başarısızlık sayacı, limit değerle karşılaştırılır. Başarısızlık sayacı limite ulaşan çözümler için, diğer bir ifadeyle,

limit sayısı kadar çevrimde daha başarılı çözümler türetemeyen çözümler yerine, Eşitlik 1 ile rastgele çözümler üretilir. Yeni oluşturulan çözümün başarısızlık sayacı sıfırlanır.

2.3. Algoritma Adımları ve Akış Diyagramı

Önceki bölümde ana işlem modülleri verilen YAK metodu, bu bölümde algoritmik tarzda işlem adımlarıyla açıklanacak ve akış şeması oluşturulacaktır.

2.3.1. YAK Algoritması Adımları

Kolay anlaşılabilir olması açısından, algoritma adımları standart sözde-kod yaklaşımıyla değil, açıklayıcı ifadeler içeren metinler eklenerek oluşturulmuştur. Yinelenen hesaplamalar ve karşılaştırma işlemleri iç satırlarda yazılmıştır.

// Başlangıç aşaması;

- **Eşitlik 1** kullanılarak ilk çözümlerin oluşturulması
- Başlangıç çözümlerinin başarısızlık değerlerinin sıfırlanması ($başarısızlık_a = 0$)
- Başlangıç çözümlerine ait amaç fonksiyonu değerlerinin ($f(\zeta_a)$) hesaplanması
- **Eşitlik 2** kullanılarak amaç fonksiyonu değerlerine göre uygunluk değerlerinin ($fit(\zeta_a)$) belirlenmesi
- *Tekrarla*
- // İşçi arı safhası
- $a=1$ 'den ζS 'ye kadar
- Eşitlik 3 ile sıradaki ζ_a ve rastgele seçilen ζ_k çözümleri kullanılarak T_a çözümünün türetilmesi
- T_a 'ya ait ($f(T_a)$) değerinin hesaplanması ve 3) ile uygunluk değerinin ($fit(T_a)$) belirlenmesi
- Eğer $fit(T_a) > fit(\zeta_a)$ ise
- Geçerli çözüm = T_a
- $başarısızlık_a = 0$
- *Aksi Halde*
- $başarısızlık_a = başarısızlık_a + 1$
- **Eşitlik 4** ile kaynakların gözcü arılarca seçilme olasılığının bulunması (r_a)

// Gözcü arı safhası;

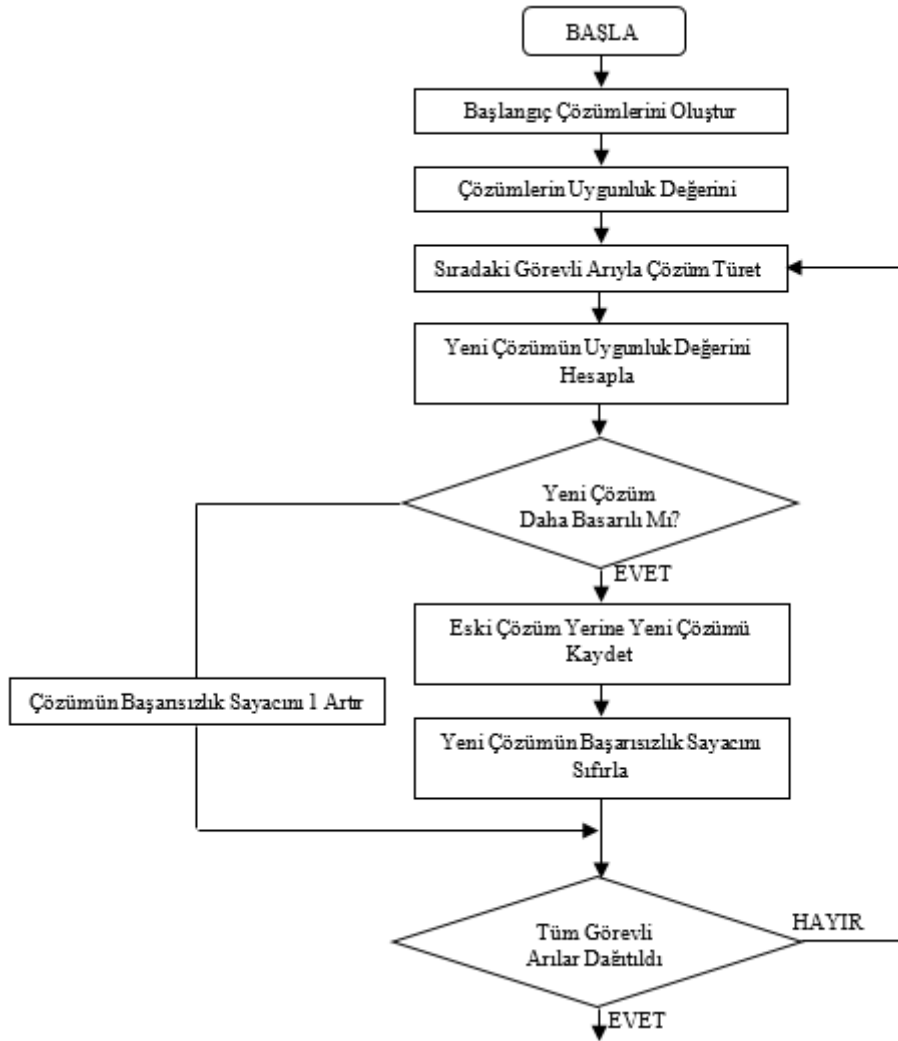
- $t=0, a=0$
- *Tekrarla*
- Eğer $rastgele_sayı < r_a$ ise
- **Eşitlik 3** ile ζ_a çözümünden T_a çözümünün türetilmesi
- T_a 'nın fonksiyon değerinin ($f(T_a)$) ve 4) ile uygunluk verisinin ($fit(T_a)$) belirlenmesi
- Eğer $fit(T_a) > fit(\zeta_a)$ ise
- Geçerli çözüm = T_a
- $başarısızlık_a = 0$

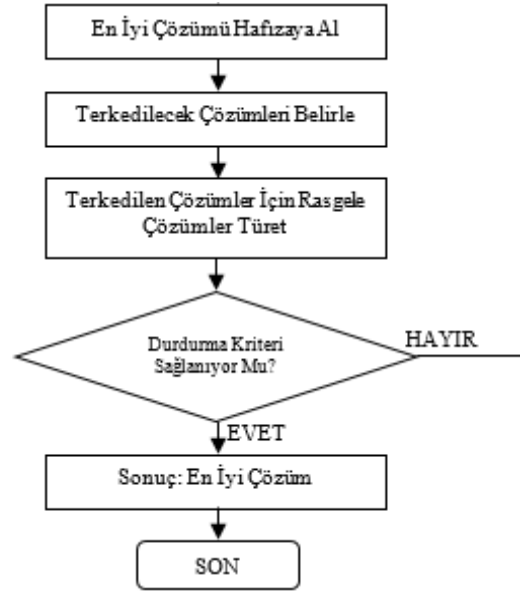
FEROMONAL YAPAY ARI KOLONİ ALGORİTMASI (FYAK)

- *Aksi Halde*
- $başarısızlık_a = başarısızlık_a + 1$
- $t=t+1$
- *Ta ki ($t=ÇS$)*
// Limit kontrolü ve kâşif arı yaklaşımı;
- *Eğer $başarısızlık_a > limit$ ise*
- $Ç_a$ silinip yerine denklem 1) ile rastgele yeni bir çözüm inşa edilmesi
- O ana kadar bulunan genel en iyi çözümün kaydedilmesi
- *Ta ki (Sonlandırma koşulları sağlanıncaya kadar)*

2.3.1.1 YAK Algoritması Akış Diyagramı

YAK algoritmasının optimizasyon problem çözümleri için oluşturulan akış şeması Şekil 3'te gösterilmektedir.

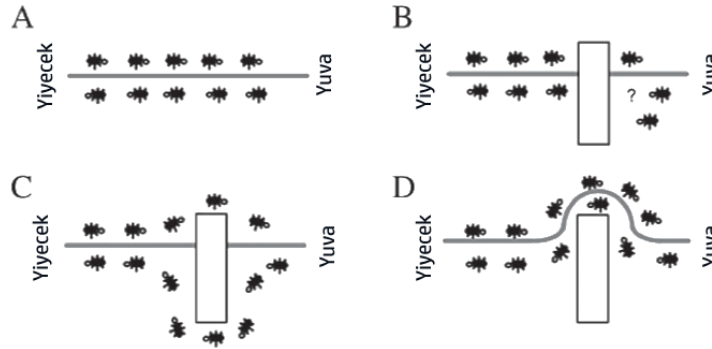




Şekil 3: YAK akış şeması

3. KARINCA KOLONİ OPTİMİZASYONU

Koloni yaşamının diğer bir meşhur örneği olan karıncalar, aralarındaki iletişimde feromon salgısını kullanırlar. Feromon, birçok mikroorganizma türü tarafından yayılan, koku özelliği olan uçucu bir salgıdır. Bu salgı, karıncalar tarafından yeni bir yiyecek kaynağı bulunduğunda ya da yuva ile kaynak arasındaki en kısa yolun belirlenmesinde aktif olarak kullanılır. Koloni üyelerince takip edilen yol, herhangi bir dış sebepten ötürü, artık en kısa yol değilse, karıncalar, değişen şartlara kolayca uyum sağlayıp, yeni en kısa yolu kısa sürede bulabilmektedirler. Şekil 4'te, besin kaynağından yuvalarına giden en kısa yolu kullanan karıncaların, önlerine engel konulduğunda, en kısa yolu yeniden nasıl bulduklarını gösterilmektedir.



Şekil 4: Önüne engel konulan karıncaların yeniden en kısa yolu bulması

Karıncaların, yiyecek bulma ve taşıma sürecindeki feromona dayanalı bu etkileşimi, optimizasyon alanındaki araştırmacıların ilgisini çekmiş ve süreç algoritmik olarak modellenmiştir (Zhang ve diğ., 2008). Bu bağlamda, temel bileşeni feromon olan Karınca Koloni Optimizasyonu (KKO) kapsamında çok sayıda karınca algoritması, literatüre kazandırılmıştır. KKO algoritmalarını birbirinden ayıran ana unsurlar: feromon güncelleme ve feromonu kullanarak yeni çözümler türetmede kullanılan geçiş kurallarıdır. Yaklaşımın, modellenen ilk yöntemi olarak Karınca Sistemi (KS) algoritması (Dorigo ve diğ., 1996), gezgin satıcı problemi (GSP) ve kuadratik atama problemleri (KAP) gibi ayrık yapıları optimizasyon problemlerine uygulanmıştır. Yöntemin performansı, süreç içerisindeki güncellemelerle iyileştirilmiş ve Karınca Koloni Sistemi (KKS) algoritması adıyla, optimizasyonun daha başarılı algoritması oluşturulmuştur (Kefayat ve diğ., 2015). fYAK kapsamında,

feromon güncelleme ve geçiş kurallarında uygulanan adımlar, KKS bazlı geliştirildiği için, çalışma kapsamında KKS algoritması açıklanacaktır.

3.1. Karınca Koloni Sistemi

Karınca koloni sistemi algoritması, Dorigo tarafından geliştirilen KKO algoritmalarından biridir (Dorigo & Gambardella, 1997). KKS de diğer KKO algoritmaları gibi özellikle kombinatoriyal optimizasyon problemlerinde çok başarılı çözümler oluşturabilmektedir. Algoritma, geçiş kuralları ve feromon güncelleme tekniği itibarıyla, KS algoritmasının tecrübelerinden yararlanmış ve geliştirilmiş modeli olarak sunulmuştur.

3.1.1. Geçiş Kuralları

KKS algoritması ile çözümler inşa edilirken, bir sonraki düğüm seçiminde iki ihtimal söz konusudur. Hangi seçim yönteminin uygulanacağı, $[0,1]$ aralığında rastgele seçilen q değeri ile belirlenir. Seçilen q değeri, q_0 'dan büyükse ikinci alternatif, aksi halde ilk alternatif uygulanır. q_0 parametresi için genellikle 0,9 gibi bir değer atandığı için, ilk alternatifi kullanma olasılığı %90 dolaylarındadır. Bu yaklaşıma göre, i düğümünde bulunan bir k karıncası, gidebileceği muhtemel düğümler (u) için feromon miktarına bağlı olarak seçilme olasılıklarını hesaplar ve seçilme olasılığı en yüksek düğümü ziyaret eder. İlk alternatifte seçilecek düğüm Eşitlik 5 ile belirlenir.

$$j = \max_{u \in j_k} \{ [\tau(i, u)]^\alpha \cdot [\eta(i, u)]^\beta \} \quad (5)$$

Burada u , i düğümünden ziyaret edilebilecek muhtemel düğümleri; $\tau(i, u)$, düğümler arasındaki feromon miktarını; $\eta(i, u)$ ise düğümler arası mesafenin tersini simgeler. $i-j$ düğümleri arası mesafe $\delta(i, j)$ ise, $\eta(i, u) = 1/\delta(i, j)$ olarak hesaplanır. α ve β parametreleri, feromon miktarının ve düğümler arası mesafenin önemini belirleyen sezgisel parametrelerdir.

İkinci alternatifte, ziyaret edilecek bir sonraki düğüm, hesaplanan seçilme olasılıklarına göre rastgele seçilir. Her bir düğümün seçilme olasılığı (p) Eşitlik 6 ile belirlenir.

$$p(i, j) = \frac{[\tau(i, j)]^\alpha \cdot [\eta(i, j)]^\beta}{\sum_{u \in j_k} [\tau(i, u)]^\alpha \cdot [\eta(i, u)]^\beta} \quad (6)$$

Burada j , ziyaret edilebilecek muhtemel u düğümlerinden biridir. Bu bağlamda ilgili düğümün seçilme olasılığı sunulmaktadır.

3.1.2. Feromon Güncelleme

Turunu tamamlayan her bir karınca yuvaya döndükten sonra, ziyaret edilen düğümler arasındaki feromon miktarları yeniden güncellenir. KKS algoritmasında feromon biriktirme işlemi iki şekilde olur: yerel feromon güncelleme, genel feromon güncelleme. Yerel feromon güncelleme işlemlerinde öncelikle, tüm düğümler arasındaki feromon değerleri, belirlenen buharlaşma oranı (ρ) kadar azaltılır. Bu işlem için Eşitlik 7 kullanılmaktadır.

$$\tau(i, j) = (1 - \rho) \cdot \tau(i, j) \quad (7)$$

Ardından tüm karıncalar için ziyaret edilen düğümlerdeki feromon miktarları, ilgili karıncanın seyahat ettiği toplam yol mesafesiyle ters orantılı olarak artırılır. m , toplam karınca sayısını temsil etmek üzere; k karıncasının uğradığı düğümlerdeki feromon artışı için Eşitlik 8 kullanılır.

$$\tau(i, j) = \tau(i, j) + \sum_{k=1}^m \Delta\tau(i, j) \quad (8)$$

FEROMONAL YAPAY ARI KOLONİ ALGORİTMASI (fYAK)

Burada $\tau(i,j)$, $i-j$ düğümleri arasındaki feromon miktarı; $\Delta\tau(i,j)$, $i-j$ düğümleri arasındaki kenarı kullanan karıncanın gittiği toplam mesafenin (L) tersidir. Bu değer, Eşitlik 9 ile hesaplanır.

$$\Delta\tau(i,j) = \begin{cases} \frac{1}{L_k} & k \text{ karıncası } i - j \text{ yolunu kullanmışsa} \\ 0 & \text{aksi takdirde} \end{cases} \quad 9)$$

Burada L_k , k karıncasının toplam seyahat uzunluğunu temsil eder.

Yerel feromon güncellemesi ile turlar dinamik olarak güncellenmiş ve bu sayede, kullanılan düğümlerin seçilme olasılığı artırılmış olur. Amaç, karıncaların daha kısa yolu bulmalarına yardımcı olmaktır.

Genel feromon güncelleme işleminde yalnızca, küresel en iyi karıncaya, diğer bir ifadeyle iterasyonun o anki çevrimine kadar en başarılı çözümü oluşturmuş karıncaya izin verilir. Buradaki amaç, bulunan en başarılı sonucun, sonraki çevrimlere taşınmasını sağlamaktır. Genel feromon güncellemede Eşitlik 10 kullanılır.

$$\tau(i,j) = (1 - \alpha) \cdot \tau(i,j) + \alpha \cdot \Delta\tau(i,j) \quad 10)$$

Burada α , feromon eksiltme değeri; $\Delta\tau(i,j)$ ise, en başarılı karıncanın gittiği toplam yol mesafesidir. $\Delta\tau(i,j)$ değeri, 11) ile belirlenir.

$$\Delta\tau(i,j) = \begin{cases} \frac{1}{L_{en\ iyi}} & \text{en başarılı karınca } i - j \text{ yolunu kullanmışsa} \\ 0 & \text{aksi takdirde} \end{cases} \quad 11)$$

4. FEROMONAL YAPAY ARI KOLONİ (fYAK) ALGORİTMASI

YAK algoritması, tasarım itibarıyla, sayısal optimizasyon problem çözümleri için hazırlanmış ve diğer birçok meta-sezgisel yöntem gibi, çözümleri, diziler halinde ele alarak optimal çözüm dizisini bulmaya çalışan bir yöntemdir. Bu sayede, arama alanının farklı bölgelerine dağılılabılır. Ancak yerel arama için hem işçi ve hem de gözcü arı parametrelerini kullanıyor olsa da istenen sömürü performansına ulaşamamıştır. Karınca algoritmaları ise, çözümleri bütün olarak yorumlamaktan ziyade, çözüm bileşenleri arasındaki korelasyonu anlamaya çalışır. Düğümler arası bağlantıyı, ilgili düğümleri kullanan çözüm sonucunu yorumlayarak öğrenmeyi hedefler. Bu bağlamda en büyük ipucu feromon bilgisidir. Feromon bilgisinin iki temel amacı vardır: çözüm bileşenleri arasındaki bağlantıyı, çözümün kalitesiyle ilişkilendirebilmek ve algoritmanın arama alanını başarılı çözüm bölgelerine yönlendirebilmek. Sonraki çevrimlerde çözümler inşa ederken, birbiriyle daha çok bağlantılı elemanları sıralamaya çalışır. Karınca algoritmaları bu yönüyle daha başarılı bir yerel arama ve sömürü yeteneğine sahiptir. Ancak yaklaşım, karıncaları en iyi çözümler etrafında biriktirme eğiliminde olduğundan, arama alanını yeterince keşfedemez.

YAK algoritmasının keşif yeteneği ile KKS algoritmasının sömürü yetenekleri birleştirilerek fYAK modeli geliştirilmiştir. Model, işçi arılar ile gözcü arılar arasındaki bilgi paylaşımını daha yüksek seviyelere çıkarmayı amaçlar. fYAK kabulünde, bal arıları en kaliteli balı üretebilmek için en kaliteli besini tüketmeye çalışır. Bu kapsamda en kaliteli besin, çiçeklerden toplanan polenlerden oluşur. Çiçeklerden polen toplayan işçi arılar, bu süreçte çiçekler arasında feromon bırakırlar. Tüm işçi arılar görevlerini tamamladıktan sonra, gözcü arılar, polen toplama sürecinde çiçekler arasındaki feromon yoğunluğunu dikkate alarak çiçekleri dolaşırlar. Bu bağlamda fYAK, yaklaşım itibarıyla standart YAK'tan şu noktalarda ayrılır:

- YAK algoritmasında bal arıları en kaliteli besin kaynağını bulmaya çalışırken, fYAK'ta arılar en kaliteli besini oluşturmaya çalışır.

FEROMONAL YAPAY ARI KOLONİ ALGORİTMASI (fYAK)

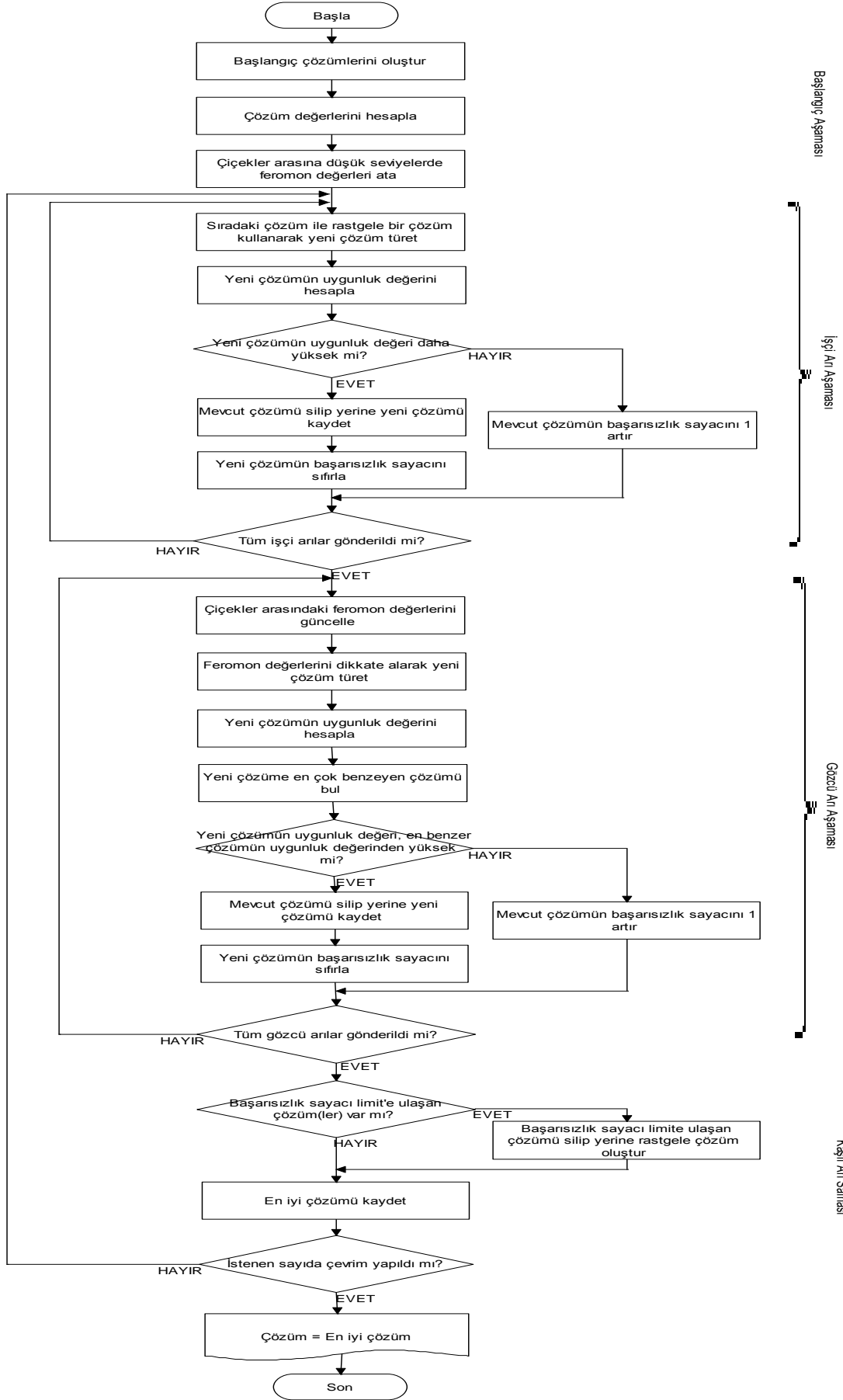
- YAK modelinde, işçi ve gözcü arılar haberleşirken titreşim hareketini kullanırken, fYAK kapsamında iletişim, feromon bilgisiyle sağlanır.
- YAK'ta gözcü arılar, daha başarılı işçi arıları takip etmeye çalışırken, fYAK'ta görevli gözcü arılar, birbiriyle daha çok ilişkili çözüm bileşenlerini sıralamaya çalışır.

YAK ile fYAK arasındaki mantık farkı, algoritmik boyutta rulet tekerleği yerine feromon değerlerinin kullanımıyla kendini gösterir. fYAK, ayrık yapılı optimizasyon problemlerinde sabit bir feromon matrisi kullanır ve matris değerleri her bir çevrimde yenilenir. Sayısal optimizasyon problemlerinde ise her bir çevrimde yeniden oluşturulan feromon matrisi vardır.

4.1. Kombinatorial Optimizasyon Problem Çözümleri İçin fYAK Modeli

KKO kapsamında feromon yaklaşımı, gerçek karıncalara sağladığı avantajlara uygun olarak modellenmiş, özellikle ayrık yapılı optimizasyon problemlerinde, algoritmalar, başarılı çözümler inşa etmişlerdir. Ancak KKO algoritmaları, en iyi çözümü başarılı çözümlerin komşuluğunda arar ve arama sürecinde yerel en iyi çözümü kolayca bulabilirler. Ancak arama alanının farklı bölgelerine yönelmekte zorlanırlar. $q0$ parametresinin değeri 1'e yaklaştığında algoritmanın sömürü yeteneği daha da artırılırken, keşif kabiliyeti azalır. Parametre değeri 0'a yaklaştığında ise durum tersine döner. Geliştirilen birçok modelde de aynı sonuçlar gözlemlenmiştir. Feromon yaklaşımı YAK gibi başarılı keşif yeteneğine sahip bir algoritmaya entegre edilerek sorun aşılabilmektedir. fYAK'ta feromon bilgisi, işçi ve gözcü arıların tecrübelerini paylaşmak üzere, keşif arılarca kullanılmaz. Algoritmanın keşif yeteneği korunarak, daha başarılı sömürü için işçi ve gözcüler arasındaki bilgi paylaşımı için kullanılır. Ayrık yapılı problemler için geliştirilen fYAK modeli Şekil 5'te sunulmaktadır (Ekmekci, 2019a).

FEROMONAL YAPAY ARI KOLONİ ALGORİTMASI (FYAK)

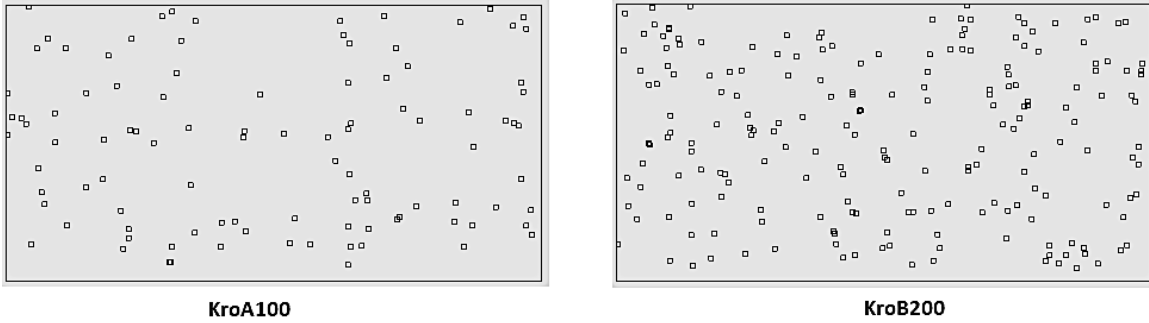


Şekil 5: Ayrık yapıli optimizasyon problemleri için FYAK akış şeması

4.1.1. fYAK'm Kombinatoriyal Optimizasyon Problemlerindeki Performansı

fYAK algoritmasının kombinatoriyal optimizasyon problem çözümlerindeki başarısını, çözüm arama stratejisini ve yakınsama performansını analiz etmek için, .NET ortamında C# programlama dili kullanılarak uygulama geliştirilmiş ve uygulama GSP örnekleri üzerinde test edilmiştir. GSP, yöneylem araştırmacılarının farklı tekniklerle çözüm aradıkları popüler örneklerden biridir. Problem için ilk matematiksel model, 1800'lü yılların sonlarında William Hamilton ve Thomas Penyntgon Kırkman tarafından tanımlanmıştır. Problemden aralarındaki uzaklıklar bilinen n adet şehir bulunur. Bu ağ modelinde amaç, şehirlerin her birine "mutlaka ve yalnızca" bir defa uğrayarak başlangıç şehrine dönmektir. $G(N, E)$ kapalı grafında N , i 'den n 'e kadar şehirleri ve E , bu şehirler arası kenarları temsil eder. Verilen C mesafe matrisinde, c_{ij} , i - j düğümleri arası mesafeyi temsil eder. Her iki düğüm arası gidiş-dönüş mesafesi eşit uzaklıktaysa ($c_{ij}=c_{ji} \quad \forall i, j \in N$) problem simetriktir, aksi halde asimetrik problem olarak adlandırılır. İki düğüm arası mesafeyi hesaplamak için çoğu zaman Öklid Formülü kullanılır.

GSP test problemleri olarak 100 ve 200 şehirden oluşan sırasıyla KroA100 ve KroB200 örnekleri seçilmiştir. Seçilen örnekler TSPLIB kütüphanesinden (<http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsp/>) indirilebilmektedir. Örneklerde şehirlerin yerleşimi Şekil 6'da gösterilmektedir. fYAK algoritmasıyla elde edilen sonuçlar, standart KKO ve standart YAK algoritmalarıyla elde edilen sonuçlarla karşılaştırılmıştır.



Şekil 6: KroA100 ve KroB200 örneklerinde şehirlerin konumu

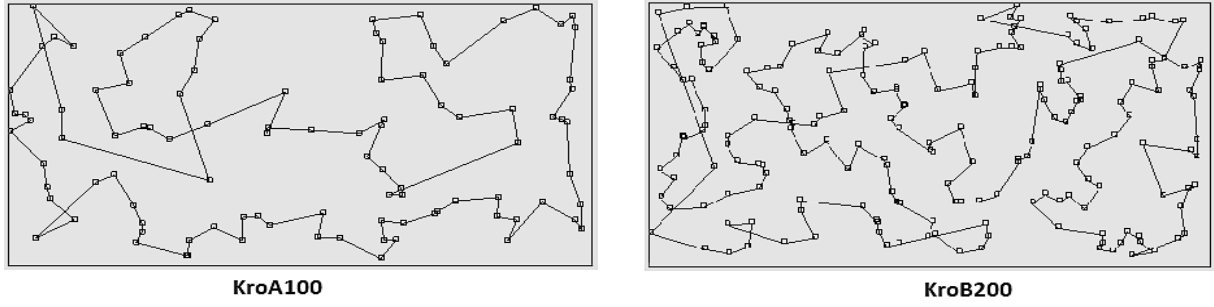
Test problemlerini çözerken fYAK, KKO ve YAK algoritmalarına ait parametreler için Tablo 1'de verilen değerler kullanılmıştır. Algoritmaların koloni boyutu parametreleri için belirlenen değerler, problemlerdeki şehir sayıları (n) ile orantılıdır.

Tablo 1: KroA100 ve KroB200 örnekleri için YAK, KKO ve fYAK için parametre seti

Algoritma	Koloni Boyutu	Limit	α	β	ρ	$q0$
YAK	n	1000	-	-	-	-
KKO	n	1000	1	5	0.1	0.9
fYAK	n	1000	1	5	0.1	0.9

Belirlenen parametre değerleriyle KroA100 ve KroB200 örnek problemlerinden 100 iterasyon sonucunda elde edilen çözümler Şekil 7'de gösterilmektedir.

FEROMONAL YAPAY ARI KOLONİ ALGORİTMASI (fYAK)



Şekil 7: fYAK ile KroA100 ve KroB200 problemleri için çözüm örnekleri

Algoritmalar, KroA100 ve KroB200 örnek problemleri için, belirlenen parametre değerlerinde toplam 5000 çevrimle, bağımsız olarak 30 kez çalıştırılmıştır. Elde edilen en iyi sonuç, en kötü sonuç, sonuçların ortalaması ve standart sapma verileri Tablo 2’de sunulmaktadır. En başarılı sonuç değerleri kırmızı renkte gösterilmektedir.

Tablo 2: KroA100 ve KroB200 çözümlerinde algoritmalarla elde edilen sonuçlar

	KroA100			KroB200		
	YAK	KKO	fYAK	YAK	KKO	fYAK
<i>En İyi</i>	24008,90	21410,31	21608,90	40240,15	33478,27	32829,05
<i>En Kötü</i>	25585,16	21911,41	21930,08	54562,00	34352,40	34115,08
<i>Ortalama</i>	24691,55	23292,12	23246,09	46303,53	33946,02	33651,32
<i>Standart Sapma</i>	817,79	736,78	784,33	3746,06	272,26	290,38

Tablo 2 verilerinden de görülebildiği gibi, feromon yaklaşımı ile YAK daha başarılı sonuçlar elde edebilmektedir. fYAK ile elde edilen tüm sonuçlar, standart YAK algoritması ile elde edilen sonuçlardan daha başarılıdır. fYAK, KroA100 probleminde KKO algoritmasına yaklaşık sonuçlar alabilmişken, daha büyük hacimli problemlerde KKO’dan daha başarılı sonuçlar elde edebilmiştir.

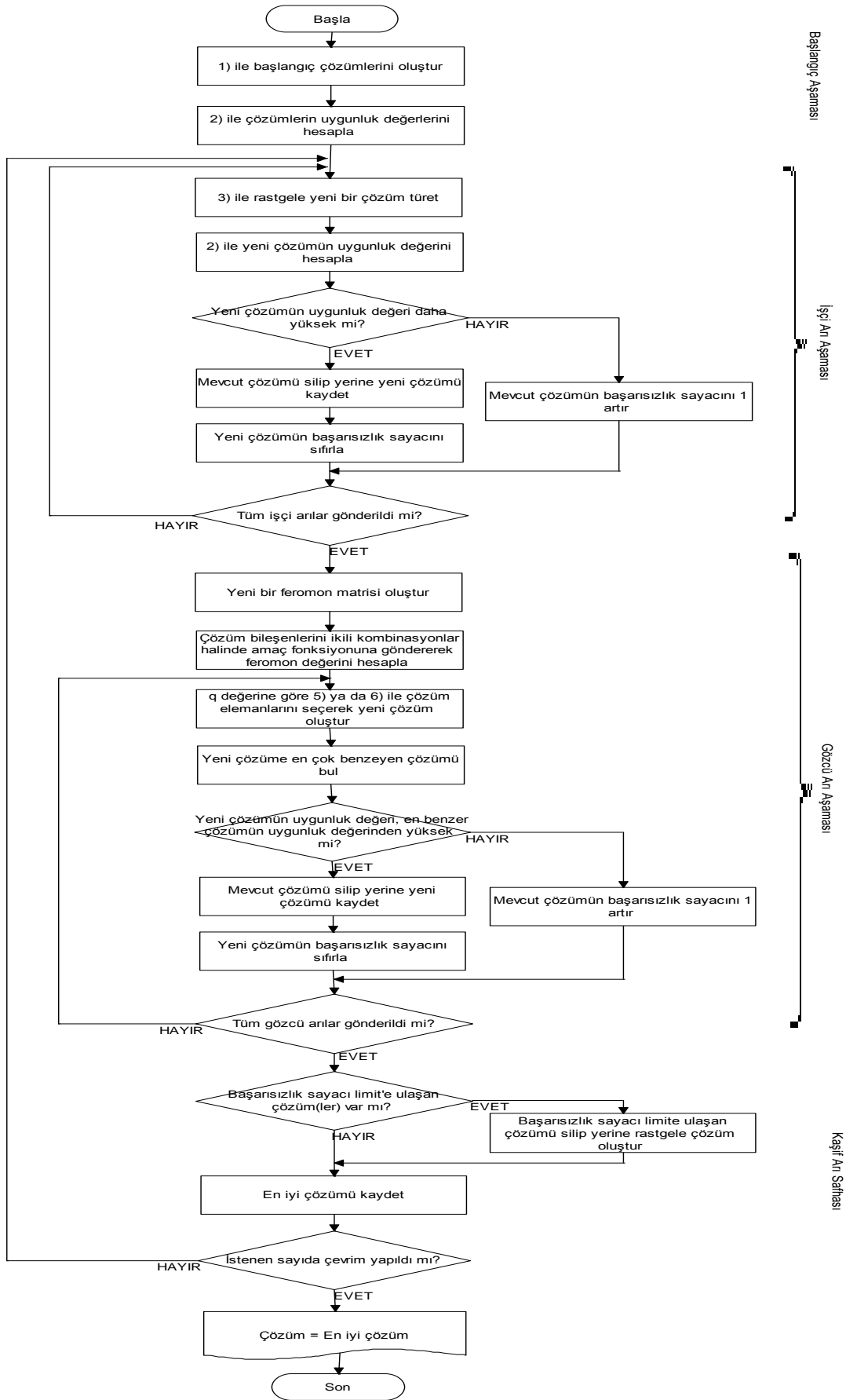
4.2. Sayısal Optimizasyon Problem Çözümleri İçin fYAK Modeli

Sayısal optimizasyon problem çözümleri, ayrık optimizasyon problem çözümlerinden farklı bir yaklaşım gerektirir (Ekmekci, 2019b). Standart YAK algoritması, sayısal optimizasyon problem çözümleri için uygundur. Ancak KKO yaklaşımı, ayrık yapıları problemler üzerinde etkilidir. Önceki bölümde de bahsedildiği üzere fYAK modelinde feromon, işçi ve gözcü arılar arasındaki bilgi alışverişini güçlendirmek; gözcü arıların, işçi arı safhasındaki tecrübeden daha çok yararlanmasını sağlamak için kullanılmaktadır.

Klasik YAK modelinde gözcü arılar, mevcut çözümlerden daha başarılı olanları seçer ve 3) eşitliğinde verildiği gibi, çözüm bileşenlerindeki değerleri, belirlenen sınırlarda rastsal olarak değiştirerek yeni çözümler üretir. fYAK modelinde ise gözcüler, çözümlerdeki sayısal değerleri nesnelere gibi yorumlayarak, sayılar arasındaki yakınlığı bulmaya çalışır. Bunun için feromon bilgisine ihtiyaç duyar. Sayılar arasına atanacak feromon değerleri ise, problemin uygunluk fonksiyonundan elde edilen sonuca göre hesaplanır. Kombinatoriyal optimizasyon problemlerinde nesnelere sabittir. Bu bağlamda, başlangıçta oluşturulan feromon matrisi her bir çevrimde güncellenir. Sayısal optimizasyon problemlerinde ise, çözüm bileşeni olan sayısal değerler her bir çevrimde değişir. Dolayısıyla işçi arı safhası tamamlandıktan sonra feromon matrisi yeniden oluşturulmaktadır.

fYAK algoritmasının sayısal optimizasyon problemleri için oluşturulan adımları Şekil 8’de gösterilmektedir.

FEROMONAL YAPAY ARI KOLONİ ALGORİTMASI (FYAK)



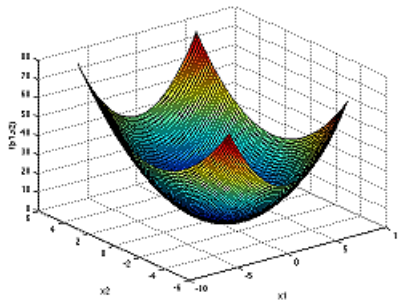
Şekil 8: Sürekli optimizasyon problemleri için FYAK akış şeması

4.2.1. fYAK'm Sürekli Optimizasyon Problemlerindeki Performansı

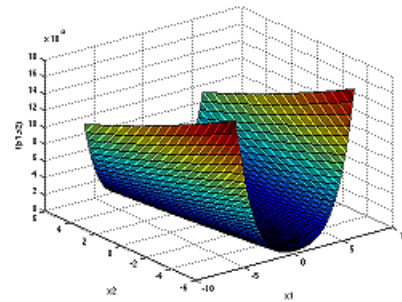
fYAK algoritmasının sürekli optimizasyon problemlerindeki performansını analiz etmek için, geliştirilen uygulama, farklı karakteristik özelliklere sahip optimizasyon fonksiyonları üzerinde test edilmiştir. Seçilen optimizasyon fonksiyonlarına ait detaylar Tablo 3'te gösterilmektedir. Çözümler oluşturulurken, çözüm bileşenleri için değerler belirlenen sınır aralıklarında ve her bir çözümde boyut sayısı kadar bileşen oluşturulmuştur. Örnek optimizasyon fonksiyonlarının, belirlenen sınır aralıklarındaki çizimleri Şekil 9'da gösterilmektedir.

Tablo 3: KroA100 ve KroB200 örnekleri için YAK, KKO ve fYAK için parametre seti

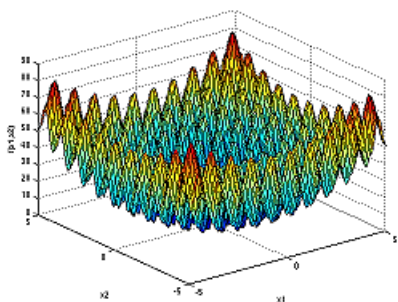
Adı	Fonksiyon	Karakteristik	Sınır Aralığı	Boyut	En İyi Sonuç
<i>Sphere</i>	$f(x) = \sum_{i=1}^D x_i^2$	Tek noktalı, Ayrılabilir	[-100, 100]	30	$F_{min} = 0, X=(0, 0, \dots)$
<i>Rosenbrock</i>	$f(x) = \sum_{i=1}^{D-1} 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2$	Tek noktalı, Ayrılmaz	[-30, 30]	30	$F_{min} = 0, X=(1, 1, \dots)$
<i>Rastrigin</i>	$f(x) = \sum_{i=1}^{D-1} 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2$	Çok noktalı, Ayrılabilir	[-5.12, 5.12]	30	$F_{min} = 0, X=(0, 0, \dots)$
<i>Griewank</i>	$f(x) = \frac{1}{4000} \left(\sum_{i=1}^D x_i^2 \right) - \left(\prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) \right) + 1$	Çok noktalı, Ayrılmaz	[-600, 600]	30	$F_{min} = 0, X=(0, 0, \dots)$



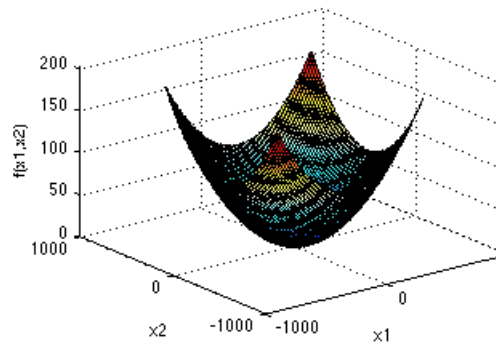
Sphere



Rosenbrock



Rastrigin



Griewank

Şekil 9: Seçilen optimizasyon fonksiyonlarının çizimi

fYAK algoritması ve YAK seçilen optimizasyon fonksiyonları bağımsız olarak 30 kez denenmiştir. Bu denemelerde, her bir çözüm dizisinde 10 adet çözüm bileşeni oluşturulmuş ve algoritmaların ortak kullandığı parametreler olarak Besin kaynağı sayısı 20, limit 200 olarak belirlenmiştir. fYAK algoritmasının feromon için kullandığı parametreler $\alpha = 1$, $\beta = 3.5$, $\rho = 0.1$ ve

$q_0 = 0.9$ olarak belirlenmiş, algoritmalarla en fazla 2000 çevrim uygulanmıştır. Algoritmaların elde ettiği sonuçlar Tablo 4'te detaylı olarak sunulmaktadır.

Tablo 4: YAK ve fYAK algoritmalarıyla elde edilen sonuçlar

Fonksiyon	Standard YAK		fYAK	
	Ortalama	Standart Sapma	Ortalama	Standart Sapma
<i>Sphere</i>	1,42E-11	1,10E-17	0	0
<i>Rosenbrock</i>	4,181065	2,071692	6,933286	2,358773
<i>Rastrigin</i>	1,07E-10	1,99E-11	0	0
<i>Griewank</i>	3,83E-19	3,16E-19	0	0

Tablo 4 verileri incelendiğinde, feromon yaklaşımının YAK algoritmasına, sürekli optimizasyon problemlerinde de olumlu bir etki oluşturduğu görülebilmektedir. fYAK, Sphere, Rastrigin ve Griewank fonksiyonlarında en iyi ortalama sonuca ulaşabildiği gibi, denemelerde alınan sonuçların standart sapma verileri de YAK'a nispeten daha düşük seviyededir. Buradan, fYAK algoritmasıyla daha kararlı sonuçlar alınabildiği yorumlanabilir.

5. SONUÇLAR

Optimizasyon problemleri, farklı bakış açılarına göre değişik şekillerde sınıflandırılabilir de temel kriter olarak, problemin çözüm bileşenleri düşünülmelidir. Örneğin, ele alınan optimizasyon problemi, birbiriyle çelişen amaçlara sahip, çok amaçlı yapıda olabilir. Bu problemin çözüm bileşenleri nesnelere oluşuyorsa, kombinatoriyal optimizasyon problemlerinde başarılı bir algoritma, bu probleme de kolaylıkla uyarlanabilir. Ancak seçilen algoritma, sayısal problemler için hazırlanmış bir meta-sezgisel ise, algoritmayı ayrık yapıya çözüme uyarlayabilmek için, hesaplamaya farklı teknikler eklenmelidir. Bu açıdan bakıldığında, çözüm elemanları 0-1'lerden oluşan problemler, yapı itibarıyla her ne kadar kombinatoriyal problemler olarak kabul edilse de sayısal optimizasyon problemlerinde daha başarılı algoritmalar, bu problemlerde daha etkili çözümler üretebilmektedir.

Bu çalışmada, farklı türden birçok optimizasyon probleminde başarılı çözümler üretebilen YAK algoritması ele alınmış ve algoritmanın yerel arama başarısını artırmak için geliştirilen fYAK modeli sunulmuştur. fYAK bir anlamda YAK ile KKS'nin birleştirildiği melez bir algoritma olarak yorumlanabilir. Literatür araştırmalarında, YAK ile KKO algoritmalarının birlikte kullanıldığı çalışmalar olduğu gibi melezleştirilerek geliştirilen tek bir algoritma da bulunabilir. fYAK, bunlardan farklı olarak, feromonu işçi ve gözcüler arasına yerleştirmektedir. Bu sayede YAK'ın keşif kabiliyeti korunmuş ve işçi arılar, gözcü arıların deneyimlerinden daha çok verim alabilmişlerdir. Çalışma kapsamında, fYAK'ın hem ayrık ve hem de sürekli optimizasyon problemleri için tasarlanan modelleri sunulmuş ve test problemlerinden elde edilen başarılar paylaşılmıştır. Tablolardaki sonuç verileri incelendiğinde, feromon yaklaşımının, standart YAK performansına olumlu etkiler kattığı görülmektedir.

fYAK halen çeşitli yapılarıdaki optimizasyon problem türlerinde, feromon parametrelerinin farklı değerleri ile denenmektedir. Bu denemelerdeki sonuçlar analiz edilerek feromon parametreleri için optimal sabit değerler belirlenebilecektir. Amaç, YAK'ın az parametre ile çözüm araştırma avantajını korumaktır.

6. KAYNAKÇA

Abbass, H. A. (2001). MBO: Marriage in Honey Bees Optimization A Haplometrosis Polygynous Swarming Approach, in *Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546)*.

Barnebau, E., Dorigo, M. & Theraulaz, G. (1999). *Swarm Intelligence From Natural to Artificial Systems*. New York: Oxford University Press.

Beni, G. & Wang, J. (1993). Swarm Intelligence in Cellular Robotic Systems, in *Robots and Biological Systems: Towards a New Bionics?*, pp. 703–712.

Dorigo, M. & Gambardella, L. M. (1997). Ant colony system: a cooperative learning approach to the traveling salesman problem, *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 53–66.

Dorigo, M., Maniezzo, V. & Coloni, A. (1991). Positive feedback as a search strategy, Milano, Italy.

Dorigo, M., Maniezzo, V. & Coloni, A. (1996). Ant system: optimization by a colony of cooperating agents, *IEEE Trans. Syst. Man Cybern. Part B*, vol. 26, no. 1, pp. 29–41.

Ekmekci, D. (2019a). A Pheromonal Artificial Bee Colony (pABC) Algorithm for Discrete Optimization Problems, *Appl. Artif. Intell.*, vol. 33, no. 11, pp. 935–950.

Ekmekci, D. (2019b). A Pheromonal Artificial Bee Colony (pABC) Algorithm for Optimization Problems, *16th International Multi-Conference on Systems, Signals & Devices (SSD2019)*, pp. 452–456.

Karaboğa, D. (2005). An Idea Based on Honey Bee Swarm for Numerical Optimization, Kayseri, Turkey.

Karaboğa, D. & Akay, B. (2007). Artificial Bee Colony (ABC) Algorithm on Training Artificial Neural Networks, in *2007 IEEE 15th Signal Processing and Communications Applications*.

Karaboğa, D. & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm, *J. Glob. Optim.*, vol. 39, no. 3, pp. 459–471.

Karaboğa, D., Gorkemli, B. & Ozturk, C. (2014). A comprehensive survey: artificial bee colony (ABC) algorithm and applications, *Artif. Intell. Rev.*, vol. 42, no. 1, pp. 21–57.

Kefayat, M., Lashkar Ara, A. & Nabavi Niaki, S. A. (2015). A hybrid of ant colony optimization and artificial bee colony algorithm for probabilistic optimal placement and sizing of distributed energy resources, *Energy Convers. Manag.*, vol. 92, pp. 149–161.

Kennedy, J. & Eberhart, R. (1995). Particle swarm optimization, in *Neural Networks, 1995. Proceedings., IEEE International Conference on*, vol. 4, pp. 1942–1948.

Pham, D. T., Ghanbarzadeh, A., Koç, E., Otri, S., Rahim, S. & Zaidi, M. (2005). *Bee Algorithm A Novel Approach to Function Optimisation*, UK, 2005.

Tereshko, V. (2000). Reaction-Diffusion Model of a Honeybee Colony's Foraging Behaviour, in *Parallel Problem Solving from Nature Lecture Notes in Computer Science, vol 1917*. Springer, Berlin, Heidelberg, pp. 807–816.

Tereshko, V., & Loengarov, A. (2005). Collective decision-making in honey bee foraging dynamics, *Comput. Inf. Syst. J.*, vol. 9, no. 3, pp. 1–7.

Yang, X. S. (2005). Engineering Optimizations via Nature-Inspired Virtual Bee Algorithms, in *Artificial Intelligence and Knowledge Engineering Applications: A Bioinspired Approach. IWINAC 2005. Lecture Notes in Computer Science, vol 3562*, pp. 317–323.

Yang, X. S. (2009). Firefly Algorithms for Multimodal Optimization, in *Stochastic Algorithms: Foundations and Applications. SAGA 2009*, vol. 5792, pp. 169–178.

Yang, X. S. & Deb, S. (2009). Cuckoo Search via Levy Flights, in *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC 2009)*, pp. 210–214.

Zhang, T., Tian, W., Zhang, Y. and Liu, S. (2008). Improved Ant Colony System for VRPSPD with Maximum Distance Constraint, *Syst. Eng. - Theory Pract.*, vol. 28, no. 1, pp. 132–140.

MAKİNE ÖĞRENME ALGORİTMALARINA GENEL BAKIŞ

Emre Avuçlu *, Abdullah Elen

** Aksaray Üniversitesi Teknik Bilimler Meslek Yüksekokulu Bilgisayar Teknolojileri Bölümü, Aksaray*

emreavuculu@aksaray.edu.tr

** Sorumlu Yazar*

1. GİRİŞ

Makine öğrenmesi, yapay zekanın bir alt dalı olarak bilinir. İnsan beyninin özelliği olan öğrenme aracılığıyla yeni bilgiler üretebilme becerisini otomatik olarak bilgisayarlar ile gerçekleştiren yöntemler makine öğrenme algoritmaları olarak bilinir. Yapay zeka, insan zekasını taklit eden sistemler veya makineler anlamına gelen geniş kapsamlı bir terimdir. Makine öğrenmesi ve yapay zeka çoğu zaman birlikte değerlendirilir. Bazı durumlarda bu iki kavram birbirinin yerine kullanılır fakat her ikisi de ayrı metodolojilerdir. Makine öğrenme algoritmalarının tümü yapay zeka sınıfının bir üyesi iken her yapay zeka algoritması bir makine öğrenme algoritması değildir. Bu çalışmada altı farklı makine öğrenme algoritmasından olan Decision Tree (Karar Ağacı), Random Forest (Rastgele Orman), K-Nearest Neighbors (K En Yakın Komşular), Support Vector Machine (Destek Vektör Makinaları), Naive Bayes ve Multinomial Logistic Regression (Çokterimli Lojistik Regresyon) algoritmaları hakkında genel bilgiler verilmiştir. Bu algoritmaların hangi amaçla hangi alanda kullanıldıkları açıklanmıştır. Algoritmalara ait kullanılan öğrenme yöntemleri açıklanmıştır. Algoritmaların birbirleriyle olan ilişkileri ve ortak ve farklı yönleri açıklanmıştır. Bu algoritmaların hangi amaçlar için hangi çalışmalarda kullanılabileceği hakkında genel bilgiler verilmiştir. Algoritmalarda kullanılan denklemlerin açıklamaları ve algoritmaları tanımlayan yöntemlerin görselleri de ayrıca açıklanmıştır. Bu çalışmanın amacı makine öğrenme algoritmaları ile ilgili kullanılan terimler hakkında temel bilgiler vermektir.

Makine öğrenmesi kavramı ilk olarak 1950 yılında Alan Turing tarafından yazılan “Bilgisayar Mekanizması ve Zeka” adlı makalesinde bilgisayarlar düşünebilir mi sorusuna dayanmaktadır. Makalesinde taklit oyunu adı verilen, bir insan, yine bir insan olan bir yargılayıcı ve bir makine olmak üzere üç katılımcıdan oluşan bir deneyi ele almıştır. Yargılayıcı bir program vasıtasıyla insan ve makineyle iletişim içindedir. Bu iletişimi sırasından insandan ve makineden almış olduğu cevaplar doğrultusunda hangisinin insan hangisinin makine olduğuna karar verir. Eğer yargılayıcı aldığı cevapların insandan mı yoksa makineden mi geldiğini sürekli olarak ayırt edemezse oyunu makine kazanmaktadır. Bu test, günümüzde Loebner Ödülleri çerçevesinde her yıl gerçekleştirilen yapay zeka yarışmasında uygulanmaktadır. Arthur Samuel makine öğrenmesini bilgisayarların açıkça programlanmadan öğrenme yeteneği kazanmasını sağlayan bir çalışma alanı olarak tanımlamaktadır. Ayrı bir alan olarak yeniden düzenlenen makine öğrenmesi 1990'lı yıllarda gelişmeye başladı. Böylece yapay zekadan miras aldığı sembolik yaklaşımlar yerine, odak noktası istatistik ve olasılık teorisine dayalı yöntemlere kaydırılmıştır [1, 3].

Dünyanın en anlaşılması zor makinesi olarak görülen insan beyni sayısal işlemleri belirli bir süre içinde yaparken kavramayla ilgili işlemleri çok kısa sürede yerine getirir. İnsan beyni herhangi bir olayla karşı karşıya kaldığı zaman o olayla ilgili durumu çok hızlı bir şekilde geçmişteki deneyimlerine dayanarak hızlı bir tepki üretir ve çözüme ulaşabilir. Bilgisayarlar ise sayısal işlemleri çok hızlı sürede yapmalarına rağmen tecrübe ile edinilmiş bilgilerin kullanımı konusunda yetersizdirler. Bu durumun sonucu olarak, acaba bilgisayarları kullanarak benzer bir zeka oluşturulabilir mi sorusunu getirmektedir [2].

Makine öğreniminin amacı, bilgisayarları belirli bir sorunu çözmek için örnek verileri veya geçmiş deneyimleri kullanacak şekilde programlamaktır [4]. Makine öğrenmesi mühendislik, bilgisayar bilimi, istatistik ve daha birçok disiplinin içinde kendine çalışma alanı bulmaktadır.

MAKİNE ÖĞRENME ALGORİTMALARINA GENEL BAKIŞ

Politikadan yerbilimlerine kadar pek çok alanda başarıyla kullanılmaktadır ve birçok problemin çözümünde kullanılmaktadır [5, 7]. Öğrenme algoritmaların kullanıldığı uygulamalara örnek verecek olursak [6]:

- Metin ve belge sınıflandırması, örnek olarak çok sayıda kişiye giden istenmeyen e-postaların tespit edilmesi,
- Doğal dil işleme, morfolojik analizler, istatistiksel ayrıştırma, metin parçası etiketleme, varlık ismi tanımlama,
- Konuşma tanıma, konuşma sentezleyici, konuşmacı doğrulama,
- Optik karakter tanıma,
- Sayısal biyoloji uygulamaları, örnek olarak protein yapı tahmini,
- Bilgisayarlı görme, örnek olarak görüntü ve yüz tanıma,
- Sahtekarlıkların saptanması (telefon ve kredi kartı), şebeke ihlali,
- Oyunlar, örnek olarak satranç ve tavlâ,
- Yardımcısız araç kontrolü (navigasyon),
- Medikal tanı,
- Tavsiye sistemleri, arama motorları, bilgi çıkarım sistemleri

İstatistik, zeka modelleri, uyumsal kontrol teorisi, psikolojik modeller, evrimsel modeller ve yapay zeka gibi disiplinler makine öğrenmesine katkıda bulunmaktadır [7].

2. MAKİNE ÖĞRENME TERMİNOLOJİSİ

2.1. Model

Bir problemin çözümüne yönelik, belirli verilerin öğrenilmesi amacıyla bir ya da daha fazla makine öğrenme algoritması kullanılarak oluşturulan bir sistemdir. Aynı zamanda hipotez olarak ta adlandırılır.

2.2. Özellik (Feature)

Verilerin kişisel olarak ölçülebilir bir özelliğidir. Sayısal özellikler kümesi, bir özellik vektörü tarafından kolayca tanımlanabilir. Özellik vektörleri, oluşturulan modelin girdilerini temsil eder. Örneğin, bir meyveyi tahmin etmek için, renk, koku, tat, vb. özellikler söylenebilir. Problemin çözümü açısından; bilgilendirici, ayırt edici ve bağımsız özellikler seçmek algoritmalar için çok önemli bir adımdır. Ham verilerden ilgili özellikleri çıkarmak için genellikle bir özellik çıkarıcı yöntem kullanılır.

2.3. Hedef (Target)

Oluşturulan model tarafından tahmin edilecek değeri temsil eder ve etiket (label) olarak ta bilinir. Özellikler bölümünde tartışılan meyve örneği için, her girdi grubuna sahip etiket, elma, portakal, muz, vb. meyvelerin adı olacaktır.

2.4. Eğitim (Training)

Fikir, bir dizi girdi (özellik) ve beklenen çıktıları (etiketler) vermektir. Bu nedenle eğitimden sonra, yeni verileri eğitilen kategorilerden birine eşleştirecek bir modelimiz (hipotezimiz) olacaktır.

2.5. Tahmin (Prediction)

Modelin eğitim süreci tamamlandığında, öngörülen çıktı (etiket) sağlayacağı bir dizi girdiyle beslenebilir.

3. MAKİNE ÖĞRENME YÖNTEMLERİ

Makine öğrenme algoritmaları; danışmanlı, danışmansız, yarı danışmanlı ve takviyeli olmak üzere dört farklı yöntem

3.1. Danışmanlı Öğrenme (Supervised Learning)

Danışmanlı öğrenme yönteminde, bir uzman tarafından oluşturulan örnek girdiler ve buna karşılık beklenen çıktılar modele sunulur. Modelin eğitim süreci, eğitim verilerinde istenen doğruluk seviyesine ulaşana kadar devam eder. Daha sonra modele yeni bir girdi sunularak tahmin edilmesi sağlanır. Bu öğrenme yöntemi, genel anlamda regresyon ve sınıflandırma olmak üzere iki alt gruba ayrılabilir. Regresyon, sürekli bir miktarı tahmin etme problemidir. BIST100'ün bir ay sonraki değeri ne olacak? Bir çocuğun yetişkin olduğundaki boy farkı ne kadar olacak? Bu yıl ülke genelindeki işsizlik oranı ne kadar olacak? Bunlar, regresyon şemsiyesi altına girebilecek sorulara örneklerdir. Sınıflandırma, sürekli miktarları tahmin etmekten ziyade, gözlemlenen girdilerin kategorilere ayrılmasıyla ilgilenir. En basit durumda, iki olası kategori vardır ve bu durum ikili sınıflandırma olarak bilinir. Örneğin bir görüntüde trafik levhası var mı? Belirli ölçüm parametrelerine göre kişinin hasta olup olmadığı gibi sorulara ikili sınıflandırma ile bir cevap bulabiliriz. Ayrıca, el yazısını tanıma gibi çoklu sınıflandırma problemlerinin çözümü için de genellikle danışmanlı öğrenme yöntemi kullanılır. Bu problemde, farklı kişiler tarafından el yazısıyla yazılmış karakterler ve bunlara karşılık gelen doğru etiketler modelin eğitiminde girdi olarak kullanılır.

3.2. Danışmansız Öğrenme (Unsupervised Learning)

Danışmansız öğrenme yönteminde, modelin öğrenme sürecinde örnek girdiler sunulur ancak, herhangi bir beklenen çıktı (etiket) verilmez. Bu nedenle, girdi verisinin hangi sınıfa ait olduğu belirsizdir ve girdi parametreleri arasındaki ilişkileri bizatihi öğrenmesi beklenir. Bu öğrenme yönteminde, ele alınan problemlerin büyük bir çoğunluğu kümeleme üzerinedir. Kümeleme, ortak bir grubun üyelerinin birbirlerine benzer ve diğer grupların üyelerinden farklı olacak şekilde gözlemlerin birlikte gruplanması anlamına gelir. Bu yöntemde yaygın olarak kullanılan bir uygulama, işletmelerde müşterilerin benzer tercihlerine, satın alma alışkanlıklarına veya demografik yapılarına göre gruplara ayrılmasıdır. Kümelemedeki en büyük zorluk, kaç kümenin olması gerektiğini veya kümelerin nasıl görünmesi gerektiğini bilmek genellikle zor veya imkansızdır.

3.3. Yarı Danışmanlı Öğrenme (Semi-supervised Learning)

Denetlenen ve denetlenmeyen öğrenme arasındadır. Çünkü eğitim için hem etiketli hem de etiketlenmemiş verileri kullanırlar – tipik olarak az miktarda etiketli veri ve büyük miktarda etiketlenmemiş veriler. Bu yöntemi kullanan sistemler öğrenme doğruluğunu önemli ölçüde iyileştirebilir. Genellikle, yarı denetimli öğrenme, elde edilen etiketli veriyi eğitmek / ondan öğrenmek için yetenekli ve ilgili kaynaklar gerektirdiğinde seçilir. Aksi takdirde, etiketlenmemiş verilerin toplanması genellikle ek kaynaklar gerektirmez.

3.4. Takviyeli Öğrenme (Reinforcement Learning)

Takviyeli öğrenme, eylemler üreterek çevresiyle etkileşime giren, olumlu ya da olumsuz durumları keşfeden bir yöntemdir. Bu yöntemde bir danışman/egitici bulunur ancak danışmanlı öğrenme modellerindeki gibi sisteme çok detay vermez veya müdahale etmez. Bunun yerine, öğrenen model bir karar verdiğinde, kararın doğru olduğu durumlar için sistemi ödüllendirir ve yanlışlar için de

cezalandırır. Bu yöntem robotik, oyun programlama, hastalık teşhisi ve fabrika otomasyonu gibi alanlarda sıklıkla kullanılır.

4. MAKİNE ÖĞRENME PROBLEMLERİ

Makine öğrenmesindeki problemlerin çözümünde matematiksel ve istatistiksel yöntemler kullanarak mevcut verilerden çıkarımlar yapan, bu çıkarımlarla bilinmeyene dair tahminlerde bulunan yöntem paradigmasıdır. Günümüzde makine öğrenmesi, geniş kapsamlı uygulamalarla birlikte bilgisayar biliminin en hızlı gelişen alanlarından birisidir. Geçmişte yapılan bazı akademik araştırmalar, makinelerin belirli problemlerin çözümünü sağlamak için bir aşamadan sonra verileri öğrenmek zorunda olduğunu göstermiştir. Bunun sonucu olarak araştırmacılar, bu konu üzerinde ortaya çıkan problemlere çeşitli sembolik yöntemlerle yaklaşımda bulunabilmek amacıyla çalışmalarını gerçekleştirmişlerdir [8]. Bu yaklaşımların önemli bir kısmı tahmin (*prediction*), kestirim (*estimation*) ve sınıflandırma (*classification*) yapabilmek becerilerine sahiptir. Öğrenen algoritmalar her geçen gün yeni uygulamalarda farklı problemlerin çözümü için kullanılmaktadır. Başlıca öğrenme problemleri aşağıdaki gibi sıralanabilir [6].

4.1. Sınıflandırma (Classification)

Sınıflandırma problemleri girdi vektörleri ve bu vektörlerin her sınıftan alınan örneklerle yapılan öğrenmelerle N kadar sınıftan hangisine ait olduğunun karar verilmesi sürecini içerir. Sınıflandırma problemindeki en önemli nokta her bir farklı örneğin tamamen bir sınıfa ait olmasıdır ve sınıfların olası tüm çıktı uzayından yola çıkmasıdır. Bu iki kısıtın gerçeğe uygun olması şart değildir, bazen örnekler kısmi olarak iki farklı sınıfa da ait olabilir.

4.2. Regresyon (Regression)

Regresyon problemlerinde çıktılar gerçek bir değer olarak tahmin edilir. Regresyon problemlerine stok değerlerinin ve ekonomik değişkenlerin değişiminin tahmini örnek olarak gösterilebilir. Böyle bir problemde yanlış bir tahminin cezası gerçek ve tahmin değerleri arasındaki farkın büyüklüğüne bağlıdır, sınıflandırma probleminin aksine farklı kategoriler arasında bir yakınlık olduğu fikri yoktur [6].

4.3. Kümeleme (Clustering)

Kümeleme, özellikle büyük veri setlerinin analizinde örneklerin farklı ölçüm yöntemleriyle birbirleri arasındaki yakınlıklarını bulup homojen bölümlenmenin yapılması işlemidir. Verilerin ve örneklerin benzerlikleri esas alınarak gruplanması olarak da tanımlanabilir [6]. Kümeleme uygulamaları gerçek hayatta birçok alanda uygulanmaktadır. Kümeleme çoğunlukla görüntü analizleri işlemlerinde, görüntü bölütlenmesinde ve sosyal ağ analizlerinde kullanılmaktadır. Bilgisayar bilimleri alanının dışında kümeleme algoritmaları, gen analizi ve suç analizinde de kullanılmaktadır [10].

4.4. Yoğunluk Tahmini (Density Estimation)

Yoğunluk Tahmini, gözlemlenen verilerin altında yatan olasılık yoğunluğu fonksiyonunu tahmin etmeyi amaçlayan istatistiksel bir alandır. İstatistiksel çıkarım, görselleştirme ve makine öğrenmede birçok görev için bir yapı taşı görevi görür. Yoğunluk Tahmini, özellikle kümelemenin uygulanması için denetimsiz öğrenme alanında yaygın olarak benimsenmiştir. En popüler ve faydalı yoğunluk tahmin tekniklerinden bazıları, Gaussian Mixtures gibi karışım modelleri ve çekirdek yoğunluğu tahmini (Kernel Density Estimation) gibi komşu temelli yaklaşımlardır. Gaussian Mixtures, danışmansız bir kümeleme şeması sağladığı için kümeleme bağlamında daha ayrıntılı olarak tartışılmaktadır.

4.5. Boyut İndirgeme (Dimensionality Reduction)

Makine öğrenmesi sınıflandırma problemlerinde, genellikle son sınıflamanın yapıldığı temelde çok fazla faktör vardır. Bu faktörler temel olarak özellikler (*features*) denilen değişkenlerdir. Özelliklerin sayısı arttıkça, eğitim setini görselleştirmek ve üzerinde çalışmak zorlaşır. Bazen, bu özelliklerin çoğu problemin çözümü için yeterli niteliğe sahip olmayabilir. Örneğin, bir hastalığın teşhisi için ölçülen 50 parametre yerine bunlardan seçilen 20 özellik ile benzer sonuçlar elde etmemiz mümkün olabilir. Bu gibi durumlarda boyut indirgeme algoritmaları devreye girmektedir. Boyut indirgeme, bir dizi ana değişken elde ederek, incelenen özellik (rastgele değişken) sayısını azaltma işlemidir.

4.6. Sıralama

Sıralama probleminin öğrenimi, son yıllarda makine öğrenmesinde çok popüler olan bir konudur. Sıralama probleminde, örnek veri setleri ele alınarak incelenir ve önem sırasına göre sıralanarak bir model oluşturulur [9]. Sıralama problemlerine örnek olarak; arama motorlarının tasarımı, bilgi çıkarımları gerektiren uygulamalar, müşteriye özgü ürün tavsiyesi gibi birçok modern uygulamada kullanılır. Veri setinin çok büyük olmasından dolayı bir sınıflandırıcı tarafından tüm parçaların konuyla ilgili olarak etiketlenmesi uygulaması zordur. Arama motorunu kullanan bir kullanıcı sorgusuna karşılık tüm dokümanlara bakmak yerine en üstteki on dokümana bakmayı tercih etmek isteyebilir. Farklı bir örnek olarak, bir kredi kartı işletmesinde sahtekarlıkların saptanmasıyla ilgili olarak çalışan bir uzmanın, bu konudaki potansiyel binlerce sınıflandırılmış işlemi incelemek yerine sadece en fazla şüpheli görülen az sayıdaki işlemi araştırması daha mantıklı olacaktır [6].

5. MAKİNE ÖĞRENME ALGORİTMALARI

Makine Öğrenmesi, matematiksel ve istatistiksel yöntemler kullanarak mevcut verilerden çıkarımlar yapan, bu çıkarımlarla bilinmeyene dair tahminlerde bulunan yöntem paradigmasıdır. Makine öğrenmesi, geniş kapsamlı uygulamalarla birlikte bilgisayar biliminin en hızlı gelişen alanlarından birisidir. Geçmişte yapılan bazı akademik araştırmalar, makinelerin belirli bir aşamadan sonra verileri öğrenmek zorunda olduğunu göstermiştir. Bunun sonucu olarak araştırmacılar, bu konu üzerinde ortaya çıkan problemlere çeşitli sembolik yöntemlerle yaklaşımda bulunabilmek amacıyla çalışmalarını gerçekleştirdiler. Bu yaklaşımların önemli bir kısmı tahmin (*prediction*), kestirim (*estimation*) ve sınıflandırma (*classification*) yapabilme becerilerine sahiptir. Bir makine öğrenmesi, sisteminin genel başarısını genelleme yeteneği olarak ta tanımlanabilir. Genelleme, bir modelin görünmeyen verileri tanımlama yeteneğini anlamına gelir. Model, eğitim setine sadece örnekleri ezberlemiyor, aynı zamanda temel veri ilişkilerini de belirleyebiliyor ve böylece eğitim verileriyle aynı dağılımı izleyen diğer verilere de genelleme yapması sonucunda bir değerlendirme işlemi yapabiliyor. Yeni verilere genelleme yapmayan model, genellikle eğitim verilerine aşırı yük getirir. Sonuçta, yeni sunulan verilerde zayıf bir performans sergiliyor. İyi bir öğrenme modeli, eğitim sırasında verilen eğitim setine aşırı uymayan ve dolayısıyla yeni verilere genelleme yapan bir modeldir [11]. Bu bölümde, literatürde sıklıkla kullanılan bazı makine öğrenme algoritmaları hakkında teknik bilgiler verilmektedir.

5.1. Decision Tree

Karar ağacı öğrenmesi (Decision Tree Learning), belirli bir kritere göre veri setinin sürekli olarak bölünmesi işlemiyle verileri sınıflandırabilen danışmanlı bir makine öğrenme algoritmasıdır. Karar ağacı öğrenmesi sırasında, öğrenilen her bilgi ağaç üzerinde modellenir. Bu ağaç yapılarında yapraklar sınıf etiketlerini, iç düğümleri birer girdiyi ve dalları ise sınıf etiketleriyle iç düğümleri arasındaki ilişkiyi temsil etmektedir. Bu yöntem veri madenciliği ve istatistik alanlarında çok sık kullanılır. Literatürde birçok karar ağacı öğrenme algoritması mevcuttur. Bunlardan yaygın olarak kullanılan yöntemlerin dört tanesi aşağıda verilmiştir.

5.1.1. ID3 Algoritması

ID3 (Iterative Dichotomiser 3) algoritması, J. Ross Quinlan [12] tarafından 1986 yılında geliştirilmiştir. Entropiye dayalı bir yöntem olup sadece kategorik veriler üzerinde çalışabilir ve açgözlü arama (greedy search) yöntemini kullanılır. Bu algortmada amaç, ağaç oluşturulurken öğrenme kümesindeki veriler mümkün olduğunca birbirine benzer hale getirilir ve böylece ağaç derinliği minimum seviyede tutulmuş olur. Bu yöntemin işlem basamakları aşağıdaki gibidir; Burada C bir eğitim kümesini temsil etsin;

- Eğer C 'deki tüm örnekler aynı sınıfın üyesi durumda ise, bu sınıf adında bir düğüm oluşturularak algoritma sonlandırılır. Aksi taktirde, v_1, v_2, \dots, v_n değerleriyle bir özellik (F) seçilir ve karar düğümü oluşturulur.
- C 'deki eğitim örnekleri, karar düğümüne (V) göre C_1, C_2, \dots, C_n olarak alt kümelere ayrılır.
- Algoritma her bir C_i kümesi için özyinelemeli olarak uygulanır.

Karar ağaçlarında bölümlenmeye hangi düğümden başlanacağı çok önemlidir. Uygun düğümden başlanmazsa ağacın içerisindeki düğümlerin ve yaprakların sayısı çok fazla olacaktır. Bu işlem için öncelikle sınıf niteliğinin entropisi hesaplanır (Eşitlik 1).

$$H(S) = -\sum_{i=1}^n p_i \log_2(p_i) \quad (1)$$

Bir sonraki aşamada, Eşitlik 2 ve Eşitlik 3'te gösterildiği gibi özellik vektörlerinin sınıfa bağımlı entropileri hesaplanır.

$$H(X_k) = -\sum_{i=1}^n \frac{|S_i|}{|X_k|} \log \frac{|S_i|}{|X_k|} \quad (2)$$

$$H(X, S) = -\sum_{k=1}^n \frac{|X_k|}{|X|} H(X_k) \quad (3)$$

Son olarak, sınıf niteliğinin entropisinden tüm özellik vektörlerinin entropisi çıkartılarak, her bir özellik için kazanç ölçütü hesaplanır. Karar ağaçlarında kazanç ölçütü olarak; bilgi kazancı (*information gain*), kazanç oranı (*gain ratio*) ve istatistikte sıklık dağılımının eşitsizlik miktarının ölçülmesinde kullanılan Gini katsayısından yararlanılabilir. Bu çalışmada kazanç ölçütü için "bilgi kazancı" yöntemi kullanılmıştır (Eşitlik 4).

$$IG(X, S) = H(S) - H(X, S) \quad (4)$$

En yüksek bilgi kazancına sahip özellik vektörü, ilgili iterasyon için dallanma düğümü olarak seçilir.

5.1.2. C4.5 Algoritması

C4.5 algoritması, Quinlan tarafından sayısal değerlere sahip nitelikler için karar ağacı oluşturma yöntemi olarak geliştirilmiştir. ID3 algoritmasından tek farkı nümerik değerlerin kategorik değerler haline dönüştürülmesidir. En büyük bilgi kazancını sağlayacak biçimde bir eşik değer belirlenir. Eşik değeri belirlemek için tüm değerler sıralanır ve ikiye bölünür. Eşik değer için $[v_i, v_{i+1}]$ aralığının orta noktası alınabilir (Eşitlik 5).

$$t_i = \frac{v_i + v_{i+1}}{2} \quad (5)$$

Böylece, nitelikteki değerler eşik değere göre iki kategoriye ayrılmış olur.

5.1.3. Twoing Algoritması

Twoing algoritmasında, eğitim veri seti her bir işlem adımında iki parçaya ayrılarak bölümlenir. Bu yöntem Gini algoritmasına göre daha dengeli ağaçlar olmasına rağmen daha yavaş çalışır. Aday bölümlerin sağ ve sol kısımlarının her biri için nitelik değerinin ilgili sütundaki tekrar sayıları alınır. Aday bölümlerin sağ ve sol kısımlarındaki her bir nitelik değeri için sınıf değerlerinin olma olasılığı hesaplanır. Son aşamada ise her bölünme için uygunluk değeri en yüksek olan alınır.

$$\Phi(B|d) = 2 \cdot \left| \frac{B_L}{T} \right| \cdot \left| \frac{B_R}{T} \right| \left| \sum_{j=1}^n \left| \frac{Tc_j}{B_L} \right| - \left| \frac{Tc_j}{B_R} \right| \right| \quad (6)$$

Burada, T eğitim verisindeki girdi sayısını, B sağ ve sol olarak aday bölünmeyi, d düğümü, Tc_j ise j . sınıf değerini temsil etmektedir.

5.1.4. Gini Algoritması

Gini algoritması ya da Gini indeksi en çok kullanılan yöntemlerden birisidir. Bu algoritmada nitelik değerleri iki parçaya ayrılarak bölümlenir. Her bölünme için Eşitlik 7 ve Eşitlik 8'deki gibi $Gini_L$ ve $Gini_R$ değerleri hesaplanır.

$$Gini_L = 1 - \sum_{i=1}^k \left(\frac{Tc_i}{B_L} \right)^2 \quad (7)$$

Burada, Tc_i soldaki bölümdeki her bir sınıf değerini, B_L sol bölümdeki tüm değer sayısını ifade etmektedir.

$$Gini_R = 1 - \sum_{i=1}^k \left(\frac{Tc_i}{B_R} \right)^2 \quad (8)$$

Burada, Tc_i sağdaki bölümdeki her bir sınıf değerini, B_R sağ bölümdeki tüm değer sayısını ifade etmektedir. Her bölümlenmeden sonra kategorilere göre Gini değerleri hesaplanır (Eşitlik 9) ve bunlardan en küçük olan seçilir.

$$Gini_j = \frac{1}{n} \left[(|B_L| \cdot Gini_L) + (|B_R| \cdot Gini_R) \right] \quad (9)$$

Burada, j veri setindeki kategori indisini, n ise veri setindeki toplam girdi sayısını temsil etmektedir.

5.2. Random Forest

Random Forest algoritması 2001 yılında Breiman [13] tarafından tasarlanmıştır. Bu yöntemde, tek bir karar ağacı üretmek yerine her biri farklı eğitim kümelerinde eğitilmiş olan çok sayıda çok değişkenli ağacın kararlarını birleştirir. Bunun sonucu olarak, sınıflandırma problemlerinin çözümünde yüksek seviyelerde başarı oranı yakalayan bir algoritmadır. Rastgele Orman algoritmasında, diğer karar ağacı yöntemlerinde olduğu gibi dallanma kriterlerinin belirlenmesi ve uygun bir budama yönteminin seçilmesi önemli bir konudur. Dallanma kriterlerinin belirlenmesinde en yaygın kullanılan kazanç ölçümleme tekniği olarak kazanç oranı (*gain ratio*) ve Gini indeksidir. Bu algoritmanın işletilmesinde, geliştirilecek olan ağaç sayısı ve her bir düğüm için kullanılan örnek sayısı olmak üzere iki farklı parametre esas alınır. Sınıflandırma sürecinde, öncelikli olarak kullanıcı

tarafından belirlenen K adet ağaç oluşturulur. Yeni bir örnek sınıflandırılacağı zaman bu K adet karar ağacı tarafından işleme tabi tutulur ve bu ağaçlardan elde edilen en yüksek orana göre yeni örneğin sınıfı belirlenir [14]. Aşağıda Random Forest algoritmasının sözde kodu gösterilmektedir.

Önkoşul: $S = (x_1, y_1), \dots, (x_n, y_n)$ bir eğitim setini, F özellikleri ve B ormandaki ağaç sayısını temsil eder.

function RastgeleOrman(S, F)

$H \leftarrow \emptyset$

for $i \in 1, 2, \dots, B$ **do**

$S^i \leftarrow$ Eğitim setinin bir bootstrap örneği

$h_i \leftarrow$ RastgeleAgacOgrenmesi(S^i, F)

$H \leftarrow H \cup \{h_i\}$

end for

return H

end function

function RastgeleAgacOgrenmesi(S, F)

Her düğüm noktasında:

$f \leftarrow F$ 'in çok küçük bir alt kümesi

f 'nin en iyi özelliğini bölümler

return Öğrenmiş ağaç

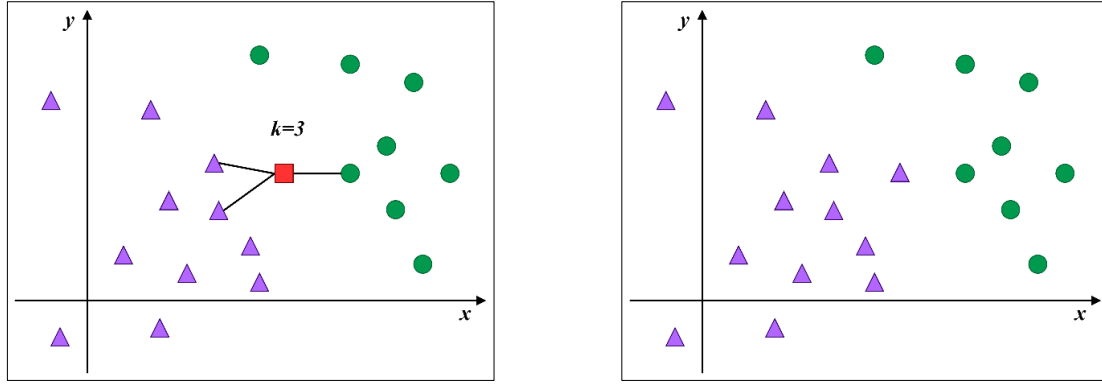
end function

5.3. K-Nearest Neighbor (k-NN)

K-En Yakın Komşu algoritması, 1967 yılında Cover ve Hart [15] tarafından önerilmiştir. k-NN, öznitelik uzayındaki en yakın eğitim örneklerine bağlı olarak nesnelere sınıflandıran, en temel örüntü tanıma ve sınıflandırma yöntemlerinden birisidir. Buradaki amaç, " k " değeri kadar en yakın komşunun ait olduğu sınıfa göre yeni örneğin hangi sınıfa ait olacağına karar vermektir. Yeni bir vektörün sınıfını belirlemek için, eğitim verisinden daha önce sınıflandırılmış olan en yakın k -adet örnek seçilir. Buna göre yeni vektör, seçilen örneklerin ait olduğu sınıflara bakılarak çoğunlukta olan sınıf hangisi ise ona atanır. Yeni bir örneğin, sınıflandırılmış olan örneklere göre mesafelerinin hesaplanması için farklı yöntemler (Öklid, Manhattan, Minkowski vb.) vardır. Bunlardan en yaygın olanı ise Öklid uzaklık hesaplama yöntemidir (Eşitlik 10).

$$d(i, j) = \sqrt{\sum_{p=1}^n (X_{ip} - X_{jp})^2} \quad (10)$$

Burada n , boyutu temsil etmektedir. i ise sınıflandırılacak olan yeni bir örneğin (X_{jp}), en yakınındaki k komşularını $X_{ip} (i = 1, 2, \dots, k)$ ifade etmektedir. Şekil 5.1'de iki boyutlu ($n = 2$) bir uzayda yeni bir X_{jp} örneğinin $k = 3$ değerine göre sınıflandırılma işlemi gösterilmektedir.



Şekil 5.1. Yeni bir örneğin ($k = 3$) değerine göre sınıflandırılması.

Şekil 5.1'deki örnekte; yeni bir örneğin veri setindeki tüm girdilere göre komşuluk mesafeleri belirlendikten sonra k komşuluk durumuna göre hangi sınıfa ait olacağı gösterilmektedir. Sonuç olarak, $k = 3$ değerine göre üçgenlerin sayıca fazla olması nedeniyle yeni örneğin sınıfı da üçgen olarak belirlenmiştir.

5.4. Support Vector Machine (SVM)

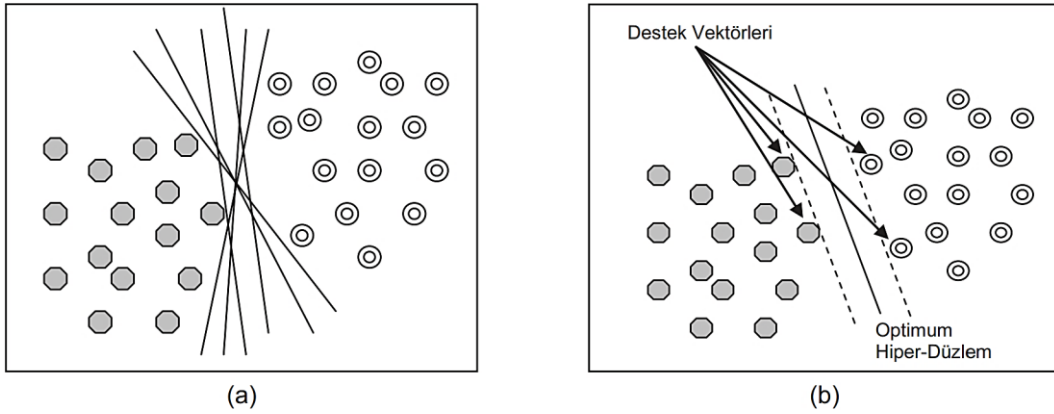
Destek Vektör Makineleri (DVM), ilk olarak 1963 yılında Vladimir N. Vapnik ve Alexey Y. Chervonenkis tarafından önerilmiştir. 1992'de Bernhard E. Boser, Isabelle M. Guyon ve Vladimir V. Vapnik çekirdek düzenlemesini (*Kernel tric*), maksimum-toleranslı çoklu düzlemlere (*hyper-plans*) uygulayarak doğrusal olmayan sınıflandırıcılar için yeni bir yöntem önermişlerdir. Verilerin doğrusal olarak ayıramadığı durumları da (*soft margin*) kapsayan mevcut yöntem, 1993 yılında Corinna Cortes ve Vapnik tarafından önerilmiş ve 1995 yılında yayınlanmıştır.

DVM, yapısal risk minimizasyonu prensibine göre çalışan ve dış bükey optimizasyona dayalı bir makine öğrenme algoritmasıdır [16]. Temel olarak ikili sınıflandırma problemlerini çözmek için tasarlanmıştır. Burada amaç, sınıfları birbirinden optimal olarak ayıracak çoklu düzleminin (*hyper-plane*) elde edilmesidir. Sınıflandırmada genellikle $\{-1, +1\}$ şeklinde sınıf etiketleri ile temsil edilir. Sınıflandırılacak olan veriler, doğrusal olarak (AND ve OR problemi) ayrılabilirdiği gibi tek bir doğruyla ayıramayan (XOR problemi) yapıda da bulunabilir. Bu nedenle DVM, verilerin durumuna göre Doğrusal DVM ve Doğrusal Olmayan DVM olmak üzere iki gruba ayrılır.

DVM, karmaşık veri setlerinde ve çözümlenmesi zor örüntülerin tanımlanmasında etkili bir öğrenme algoritmasıdır. Algoritma, önceden gözlenmemiş verilerin sınıflandırma kestirimi için örneklerden ayırt edebilen bir sınıflandırma öğrenmesini gerçekleştirmektedir. Bilindiği üzere, gerçek dünyada sınıflandırma problemlerinin büyük çoğunluğu ikiden daha fazla sınıftan oluşmaktadır. Bu türden problemlerin çözümü için çok sınıflı bir DVM sınıflandırıcısına ihtiyaç duyulmaktadır. Çoklu sınıflandırma, ikili sınıflandırıcıların birleştirilmesiyle elde edilebilir [17]. DVM sınıflandırıcısı, büyük boyutlu verileri ele alma ve farklı veri kaynaklarını yüksek doğrulukta modelleme esnekliği nedeniyle Biyoinformatikte ve diğer disiplinlerde yaygın olarak kullanılmaktadır [18].

5.4.1. Doğrusal DVM

DVM, eğitim veri setinin doğrusal olarak ayrılabilme durumunda, en geniş aralığa sahip ayırma düzlemini bulmaya çalışır. Şekil 5.2(a)'da gösterildiği üzere iki sınıflı veri setini birbirinden ayırabilen sonsuz sayıda düzlem çizilebilir. Ancak buradaki amaç, kendisine en yakın veri örnekleri arasındaki mesafeyi maksimuma çıkaran düzlemi bulabilmektir. Bu mesafeye genellikle tolerans denir ve buna karşılık gelen sınıflandırıcı, maksimum tolerans sınıflandırıcısı olarak bilinir. Şekil 5.2(b)'de görüldüğü üzere sınırı maksimuma çıkararak ayırımı yapan düzleme optimal ayırma düzlemi denir [19]. Bu ayırma düzlemi, her iki tolerans sınırına eşit uzaklıkta bulunur. Tolerans sınırları üzerinde bulunan veri örnekleri ise destek vektörleri olarak adlandırılır. Söz konusu destek vektörleri, ait olduğu sınıfın sınırını belirler ve optimal ayırma düzlemine paralel bir düzlem üzerinde yer alır [20].

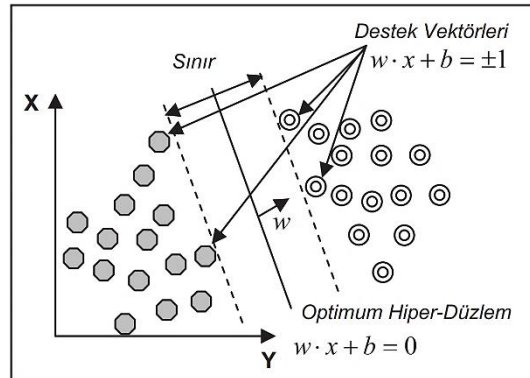


Şekil 5.2. Doğrusal DVM'lerde ayırma düzlemleri; (a) İki sınıflı bir problem için hiper-düzlemler, (b) Optimum hiper-düzlem ve destek vektörleri.

Doğrusal olarak ayrılabilen iki sınıflı bir sınıflandırma probleminde DVM'nin eğitimi için n sayıda örnekten oluşan eğitim verisinin $\{x_i, y_i\}, i = 1, 2, \dots, n, y_i \in \{-1, +1\}, x_i \in R^d$ olduğu kabul edilirse, optimal ayırma düzlemine ait karar fonksiyonu Eşitlik 11'deki gibi olur:

$$y_i = \begin{cases} w \cdot x_i + b \geq +1, & +1 \\ w \cdot x_i + b \leq -1, & -1 \end{cases} \quad (11)$$

Burada R^d , x_i girdi örüntülerinin d boyutlu uzayını, y_i girdilerin $\{-1, +1\}$ olarak sınıflandırılacağı etiketleri, w çoklu düzlemin normalini (ağırlık vektörü), b eğilim değerini (bias) temsil etmektedir. Optimal ayırma düzleminin belirlenebilmesi için bu düzleme paralel olan sınırların belirlenmesi gerekir. Yani destek vektörlerinin bulunması gerekmektedir. Bu işlem Şekil 5.3'te görüldüğü üzere $w \cdot x_i + b = \pm 1$ şeklinde ifade edilir [19].



Şekil 5.3. Doğrusal olarak ayrılabilen veri setleri.

Tolerans değerinin maksimuma çıkarılması için $\|w\|$ ifadesinin minimum hale getirilmesi gerekir. Bu durumda optimal düzlemin belirlenebilmesi için Eşitlik 12'deki gibi sınırlı optimizasyon probleminin çözülmesi gerekir.

$$\min \left[\frac{1}{2} \|w\|^2 \right] \quad (12)$$

Buna bağlı sınırlamalar ise $y_i(w \cdot x_i + b) - 1 \geq 0$ şeklinde ifade edilir. Bu optimizasyon problemi Lagrange denklemleri kullanılarak çözülebilir. Bu işlem sonrasında Eşitlik 13'deki gibi;

$$L(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^k \alpha_i y_i (w \cdot x_i + b) + \sum_{i=1}^k \alpha_i \quad (13)$$

eşitliği elde edilir. Sonuç olarak, doğrusal olarak ayrılabilen iki sınıflı bir problem için karar fonksiyonu Eşitlik 14'deki gösterildiği şekilde yazılabilir;

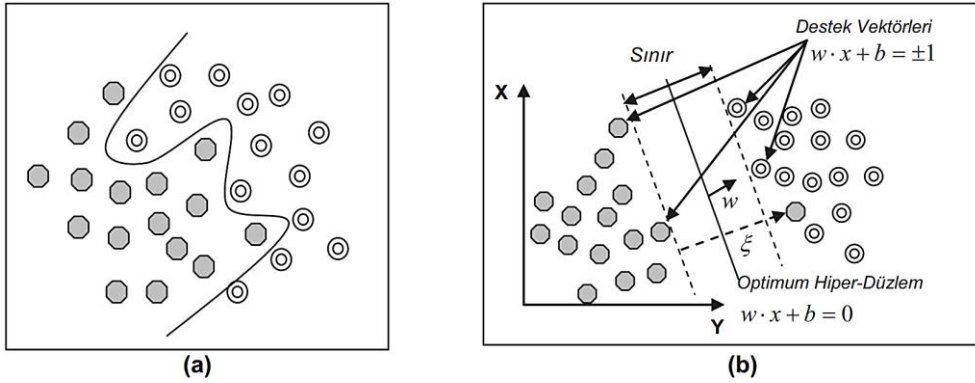
$$f(X) = \text{sign} \left(\sum_{i=1}^k \lambda_i y_i (X \cdot x_i + b) \right) \quad (14)$$

5.4.2. Doğrusal Olmayan DVM

Tıbbi görüntülerin sınıflandırılmasında olduğu gibi diğer birçok görüntü işleme probleminde de Şekil 5.4(a)'da gösterildiği şekilde verilerin doğrusal olarak ayrılması mümkün değildir. Bu durumda, eğitim verilerinin bir kısmının optimum hiper-düzlemin diğer tarafında kalmasından kaynaklanan problem pozitif bir yapay değişkenin (ξ_i) tanımlanması ile Şekil 5.4(b)'de gösterildiği gibi çözülmesi mümkün olur [19]. Sınır değerinin maksimum hale getirilmesi ve yanlış sınıflandırma hatalarının minimum hale getirilmesi arasındaki denge pozitif değerler alan ve C ile gösterilen bir düzenleme parametresi ($0 < C < \infty$) tanımlanmasıyla kontrol edilebilir [21]. Düzenleme parametresi ve yapay değişken kullanılarak doğrusal olarak ayıramayan veriler için optimizasyon problemi Eşitlik 15'deki gibi:

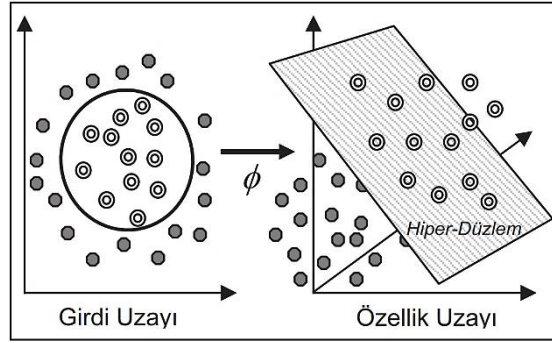
$$\min \left[\frac{1}{2} \|w\|^2 + C \sum_{i=1}^r \xi_i \right] \quad (15)$$

şeklini alır. Buna bağlı sınırlamalar ise $y_i((w, x_i) + b) - 1 \geq 1 - \xi_i$, $\xi_i \geq 0$ şeklinde ifade edilir.



Şekil 5.4. Doğrusal olmayan DVM'lerde ayırma düzlemleri; (a) Doğrusal olarak ayıramayan veri seti, (b) Doğrusal ayıramayan veri setleri için ayırma düzleminin belirlenmesi.

Eşitlik 15'te ifade edilen optimizasyon probleminin çözümü için Şekil 5.5'te görüleceği üzere girdi uzayında doğrusal olarak ayıramayan veri, özellik uzayı olarak tanımlanan yüksek boyutlu bir uzayda görüntülenir [19]. Böylece verilerin doğrusal olarak ayırımı yapılabilmekte ve sınıflar arasındaki hiper-düzlem belirlenebilmektedir.



Şekil 5.5. Çekirdek fonksiyonu kullanarak veri setinin daha büyük bir boyuta dönüştürülmesi.

DVM'leri matematiksel olarak $K(X, x_i) = \varphi(X) \cdot \varphi(x_i)$ şeklinde ifade edilen bir çekirdek fonksiyonu yardımıyla doğrusal olmayan dönüşümler yapılabilen ve bu şekilde verilerin yüksek boyutta doğrusal olarak ayırılmasına imkân sağlamaktadır. Sonuç olarak, çekirdek fonksiyonu kullanarak doğrusal olarak ayıramayan iki sınıflı bir problemin çözümü ile ilgili karar kuralı Eşitlik 16'daki şekilde yazılabilir [22]:

$$f(X) = \text{sign} \left(\sum_{i=1}^k \alpha_i y_i \varphi(X) \varphi(x_i) + b \right) \quad (16)$$

DVM'ler ile gerçekleştirilecek bir sınıflandırma işlemi için kullanılacak çekirdek fonksiyonu ve bu fonksiyona ait optimum parametrelerin belirlenmesi esastır.

5.4.3. Çekirdek Düzenlemesi ve Fonksiyonları

Denklemler dönüştürülmüş uzayda iki vektörün iç çarpımı biçimindedir. Boyut sorunundan dolayı hesaplanması zordur. Bu sorunu önlemek amacıyla "Çekirdek Düzenlemesi" önerilmiştir. Çekirdek düzenlemesi yapılarak dönüştürülmüş uzaydaki $\Phi(x)$ vektörü yerine girdi uzayındaki verilerden oluşturulan bir çekirdek fonksiyonu (Çizelge 5.1) ile işlemler yapılır. Orijinal veriyi kullanarak dönüştürülmüş uzayda bir benzerlik hesaplaması yapar.

Çizelge 5.1. DVM'de kullanılan temel çekirdek fonksiyonları ve parametreleri.

Çekirdek Fonksiyonu	Matematiksel İfadesi	Parametre
Lineer Çekirdek	$K(x, y) = xy + c$	
Polinom Kerneli	$K(x, y) = (xy + c)^d$	Polinom derecesi
Radyal Tabanlı Fonksiyon Kerneli	$K(x, y) = e^{-\gamma \ x - x_i\ ^2}$	Kernel boyutu
Pearson VII (PUK) Kerneli	$K(x, y) = \frac{1}{\left[1 + \left(\frac{2\sqrt{\ x - y\ ^2 \sqrt{2^{(1/\omega)} - 1}}}{\sigma} \right)^2 \right]^\omega}$	Pearson genişliği parametreleri
Sigmoid Çekirdek	$K(x, y) = \tanh(ax \cdot y - b)$	

Çekirdek fonksiyonuna özgü parametrelerin yanında tüm DVM için düzenleme parametresi C 'nin (tolerans) kullanıcı tarafından belirlenmesi gerekir. Bu parametre için olmasa gerekenden çok

küçük veya çok büyük değerler seçilmesi durumunda optimum hiper-düzlem doğru belirlenemeyeceğinden, sınıflandırma doğruluğunda ciddi düşüş beklenmektedir.

5.4.4. Çok Sınıflı DVM

DVM’de, ikiden fazla sınıf olması durumunda, ele alınan problemin çözümü için üç farklı yaklaşımdan birisi izlenir:

1. Problemin ikili gruplara indirgenmesi (bire bir, *one-to-one*)
2. Problemin tek gruptan bütün gruplara modellenmesi (bire çok, *one-to-many*)
3. Çok sınıf sıralama DVM (*Multiclass Ranking SVM*)

Bu yaklaşımlardan en çok kullanılan “bire bir” yöntemine göre; örneğin bir girdinin A, B ve C olmak üzere üç sınıftan hangisine gireceği bulunmak istenirse; bu sınıflar öncelikle ikili gruplar halinde eğitsek sınıfların birbirine göre çarpanları çıkartılır. Çoklu sınıflandırmada kullanılan bire çok yaklaşımına (*one-to-many*) göre de problem “kazanan hepsini alır” (*Winner takes all*) haline dönüşmektedir. Bu yaklaşımda yeni gelen bir girdinin sınıflandırılacağı alternatiflerden birisi diğerine göre baskın olmakta ve sonuçta bu sınıf seçilmektedir. Ancak bu yaklaşım bir önceki yaklaşıma göre daha az tercih edilmektedir. Son olarak, çoklu sınıf sıralama (*multiclass ranking*) yaklaşımına göre sınıflar üzerinden bir DVM çalışarak, bu sınıfların bütün üyelerini içeren bir sınıflandırma yapmaya çalışır. Ancak bu yaklaşım diğer iki yaklaşımdan daha az tercih edilir olmasında bu yaklaşımda eğitim süresinin diğerlerine göre inanılmaz oranda yüksek olması ve sonucun bulunamama ihtimali önemli rol oynar.

5.5. Naive Bayes

Naive Bayes (NB) Sınıflandırıcısı, Bayes teoremine (Thomas Bayes, 1702-1761) dayanan basit bir olasılıksal sınıflandırma yöntemidir. Bayes teoreminde (BT), birbirinden bağımsız olan ve rastgele iki olayın (X ve Y) arka arkaya gerçekleştiği durumlarda, bu iki olaydan birinin gerçekleşmesi durumunda ikinci olayın gerçekleşme olasılığı $P(X \cap Y)$ ifadesi ile gösterilebilir. Değişme özelliği sayesinde Eşitlik 17’deki gibi çarpım kuralı iki farklı ifade ile yazılabilir;

$$P(X \cap Y) = P(X|Y)P(Y) = P(Y|X)P(X) \quad (17)$$

Bayes teoremi, rassal bir sürece bağlı olarak ortaya çıkan rasgele bir X olayı ile diğer bir rasgele Y olayı için koşullu olasılıklar ve marjinal olasılıklar arasındaki ilişkiyi tanımlar (Eşitlik 18).

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)} \quad (18)$$

Herhangi bir problemde gerçekleşmesi muhtemel bağımlı durumların olasılıkları yukarıda verilen Bayes eşitliği ile hesaplanır. Bu eşitlikte $P(X)$ ifadesi problemin girdi olasılığını, $P(Y)$ ifadesi olası çıkış durumunun olasılığını ve $P(Y|X)$ ifadesi ise daha önce gerçekleşen X girişine karşın Y çıkışı durumlarının olasılığını temsil etmektedir [23]. NB sınıflandırma tekniğinde, her bir ilişkiyi koşullu bir olasılık oluşturmak için bağımlı ve bağımsız özellikler arasındaki ilişkiyi analiz eder. Yeni bir örneği sınıflandırmak için bağımlı değişken üzerinde bağımsız değişkenlerin etkilerini birleştirerek bir tahmin yapılır [24]. Bu yöntemde sınıflandırma işlemi için; ilk aşamada, muhtemel çıkış durumları içerisindeki en büyük olasılığa sahip olan durum y_{MPH} (*maximum a posteriori hypothesis*) olarak belirlenir.

$$y_{MPH} = \arg \max_{y_j \in Y} P(Y = y_j | X) \quad (19)$$

Burada, y_j muhtemel j . çıkış durumunu ve X değişkeni ise sınıf ataması yapılacak olan giriş örneğini ifade etmektedir. BT kullanılarak Eşitlik 19'daki denklem şöyle yazılabilir:

$$y_{MPH} = \arg \max_{y_j \in Y} \frac{P(y_j | x_1, x_2, \dots, x_n) P(y_j)}{P(x_1, x_2, \dots, x_n)} \quad (20)$$

$$y_{MPH} = \arg \max_{y_j \in Y} P(x_1, x_2, \dots, x_n | y_j) P(y_j) \quad (21)$$

Eğitim verisine dayanarak Eşitlik 19'daki iki terim hesaplanır. Her $P(y_j)$ değeri, eğitim verisinde geçen, her hedef değer y_j 'nin frekansı sayılarak kolaylıkla hesaplanabilmektedir. Bu terimlerin sayısının muhtemel örnekler ile hedef değerleri çarpımına eşit olması probleme neden olabilir. Bu nedenle, daha reel değerler elde edebilmek için her bir örneğin, örnek uzayında defalarca gösterilmesi gerekir. Naive Bayes sınıflandırıcısı, özellik değerlerinin hedef değerden bağımsız olması kabulüne dayanır. Bir başka deyişle, (x_1, x_2, \dots, x_n) bağlamının olasılığı, herbir özelliğe ait olasılıklarının çiktısıdır [25].

Giriş örneğinin birden çok özelliğe sahip olması durumunda, Bayes formülü farklı bir biçime dönüşecektir. Birçok niteliğin keşimi görünümündeki veri örneği için hedef sınıf tahmininde tüm özellikler için Eşitlik 22'deki gibi koşullu olasılıkların çarpımı hesaplanmalıdır.

$$P(x_1, x_2, \dots, x_n | y_j) = \prod_{i=1}^n P(x_i | y_j) \quad (22)$$

Burada, $X = (x_1, x_2, \dots, x_n)$ giriş örneği olup n adet özellikten oluşmaktadır. NB sınıflandırıcı ile BT hesaplamalarında dikkat edilmesi gereken en önemli farklılık, sınıflandırıcının olasılık değerinden daha çok "hedef sınıfı" bulmaya odaklanmış olmasıdır. BT'de paydada bulunan değer, tüm hedef durumların olasılık hesaplarında ortak olduğundan NB sınıflandırıcıda ihmal edilir. Bu durumda çok özelliğe sahip girişlerde NB formülü Eşitlik 23'deki gibi olur.

$$y_{NB} = \arg \max_{y_j \in Y} P(y_j) \prod_{i=1}^n P(x_i | y_j) \quad (23)$$

Burada y_{NB} , NB sınıflandırıcısının çıktısını, yani sınıfını temsil eder. Kısaca, NB sınıflandırma yöntemi, çeşitli $P(y_j)$ ve $P(x_i|y_j)$ terimlerinin eğitim verisi üzerindeki frekanslarına dayanarak hesaplandığı bir öğrenme sürecidir [25].

5.6. Multinomial Logistic Regression

Regresyon analizi, aralarında sebep-sonuç ilişkisi bulunan iki veya daha fazla değişken arasındaki ilişkiyi belirlemek ve bu ilişkiyi kullanarak, konu ile ilgili tahminler ya da kestirimler yapabilmek amacıyla kullanılan istatistiksel bir yöntemdir. Bu yöntemde iki veya daha fazla değişken arasındaki ilişki açıklamak için matematiksel bir model kullanılır ve bu model "Regresyon Modeli" olarak adlandırılır. Bağımlı değişkenin normal dağılım gösterdiği ve sürekli bir değişken olduğu durumlarda "Doğrusal Regresyon (DR)" analizi kullanılabilir. Bağımlı değişkenin kategorik olması durumunda "Lojistik Regresyon (LR)" ve "Multinomial Lojistik Regresyon (MLR)" analizi tercih edilen yöntemlerdir [26-28]. LR, iki bağımlı değişken için tasarlanmış doğrusal olmayan bir regresyon modelidir. LR, literatürde "Logit Regresyon" olarak adlandırılmaktadır [29]. MLR ise bağımlı değişkenin en az üç veya daha fazla kategori içerdiği [30] ve değerlerinin sınıflayıcı ölçekle elde edildiği durumlarda, bağımlı değişken (Y) ile bağımsız değişkenler (X) arasındaki neden sonuç

ilişkilerini açıklamak için kullanılmaktadır [27, 28, 31]. Bu analizde amaç, kategorik olarak bağımlı değişkenlerin değerini tahmin etmek olduğundan, aslında burada yapılmaya çalışılan iki veya daha fazla kategoriye ilişkin “üyelik” tahminidir. Buna göre, amaçlarından birinin sınıflandırma, diğerinin ise bağımlı ve bağımsız değişkenler arasındaki ilişkileri araştırmak olduğu ifade edilebilir [32]. Bir diğer ifade ile, gözlemleri ait oldukları sınıflara en doğru şekilde atayacak ve gözlemlere ilişkin yapıları ve risk faktörlerini belirleyebilecek modeli kurmayı amaçlamaktadır [33]. LR’de, bir p olayının olma olasılığının kendi dışında kalan diğer olayların olma olasılıklarının oranına “odds” ya da “üstünlük” değeri denilmektedir (Eşitlik 24) ve bu oran, LR modelinin doğrusallaştırılması esnasında dönüşümü kolaylaştırıcı bir işlev vazifesi görmektedir.

$$Odds = \frac{p}{1-p} = \frac{\pi(x)}{1-\pi(x)} \quad (24)$$

LR modeli, genel doğrusal modellerin binom dağılımlı olarak bağımlı değişkenler için elde edilmiş olan özel bir biçimidir ve Eşitlik 25’deki gibi ifade edilir;

$$\pi(x) = \frac{\alpha + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_p x_p}{1 + e^{\alpha + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_p x_p}} \quad (25)$$

Burada, $\pi(x)$ incelenen bir olayın görülme olasılığını, α bağımlı değişken sabitini, $\beta_1, \beta_2, \dots, \beta_p$ bağımsız değişkenlerin regresyon katsayılarını, x_1, x_2, \dots, x_p bağımsız değişkenleri, p bağımsız değişken sayısını ve e hata terimini temsil etmektedir. MLR modeli, Eşitlik 26’da gösterildiği şekilde, iki durumlu olan LR modelinin genişletilmiş biçimidir.

$$\pi_j(x_i) = \frac{e^{\alpha_i + \beta_{1j} x_{1i} + \beta_{2j} x_{2i} + \dots + \beta_{pj} x_{pi}}}{1 + \sum_{j=1}^{k-1} e^{\alpha_i + \beta_{1j} x_{1i} + \beta_{2j} x_{2i} + \dots + \beta_{pj} x_{pi}}} \quad (26)$$

Burada, j_1, j_2, \dots, j_k olmak üzere k adet kategoriye, i_1, i_2, \dots, i_n olmak üzere n adet olası bağımsız değişken seviyelerini temsil etmektedir. Lojistik regresyon analizinin özellikle son 20 yıldır askeri konularda, meteorolojide, iç göç hareketlerinde ve eğitim alanında kullanımının arttığı görülmektedir. Bu artışın en önemli sebeplerinden birisi, istatistik ile ilgili paket programlarının kullanımının yaygınlaşmış olmasıdır. Ancak yine de daha yaygın kullanıldığı alanlardan birinin tıp olduğu görülmektedir [32]. Lojistik modelin biyolojik deneylerin analizi için kullanımı ilk olarak Berkson (1944) tarafından önerilmiş, Cox (1970) bu modeli gözden geçirerek çeşitli uygulamalarını yapmıştır [34].

6. SONUÇLAR

İnsanlık teknolojik gelişmelerle birlikte, hayatının birçok alanında bilgisayarlar sayesinde zorlu analizleri inceleyebilir ve uzun sürecek işlemleri kısa sürede çözebilir hale gelmiştir. Son zamanlarda bilgisayarlar ile gerçekleştirilen işlemler, geçtiğimiz yıllara oranla farklı bir boyut kazanmıştır. İnsan beyninin çalışma yapısı günümüzde genel hatlarıyla belirlenebilmiş olsa da belirlenen bu yapıya benzer şekilde bilgisayarların, yani makinelerin, çalışma prensiplerinin yapılandırılması amaçlanmıştır. Sonuç olarak insan doğasından kaynaklı kısıtlardan arındırılmış, fakat insan beyninin sahip olduğu karmaşık veri işleme yeteneğine sahip makine öğrenme yöntemleri geliştirilmiştir. Bu yöntemlerde, veri setinin özelliklerine ve erişilmek istenen amaca göre farklı teknikler (farklı algoritmalar) kullanılarak, karar veren modeller geliştirilmiştir. Makine öğrenmesi, gerçek hayattaki karar verme ihtiyaçlarını karşılamak için modellerinin gelişimini hedefler. Bu

doğrultuda birçok alanda başarılı bir şekilde uygulanmaktadır. Bu çalışmada, makine öğrenme algoritmaları hakkında genel bilgiler ve literatürde en çok kullanılan Karar Ağacı, Rastgele Orman, K En Yakın Komşu, Destek Vektör Makinaları, Naive Bayes ve Multinomial Lojistik Regresyon algoritmalarının çalışma prensipleri açıklanmıştır.

7. KAYNAKÇA

- [1]. Bell, J. Machine Learning: Hands-On For Developers And Technical Professionals, Kindle Edition, NY, USA, John Wiley & Sons Inc, 2014.
- [2]. Elmas, P. Yapay Zeka Uygulamaları, Ankara, Türkiye, Seçkin Yayıncılık, 2010.
- [3]. Langley, Pat, “The changing science of machine learning”, Machine Learning. 82 (3): 275–279 (2011).
- [4]. Alpaydın, E., “Introduction to Machine Learning”, London: The MIT Press, 2004.
- [5]. Harrington, P. Machine Learning In Action, NY, USA, Manning Publications Co, 2012.
- [6]. Mohri, M., Rostamizadeh, A. ve Talwalkar, A. Foundations Of Machine Learning, NY, USA, The MIT Press, 2012.
- [7]. Nilsson, N. J. Introduction To Machine Learning (An Early Draft Of A Proposed Textbook), CA, USA, 1998.
- [8]. Sarle, W. S., “Neural Networks and Statistical Models”, *Proceedings of the Nineteenth Annual SAS Users Group International Conference*, Texas, 1-13 (1994).
- [9]. Cao, Z., Qin, T., Liu, T.Y., Tsai, M.F. ve Li, H. “Learning To Rank: From Pairwise Approach To Listwise Approach”, 24th International Conference on Machine Learning, 2007, pp. 139-146.
- [10]. Wali, A. Clojure For Machine Learning, Birmingham, UK, Packt Publishing, 2014.
- [11]. H. Blockeel, 2010. Machine Learning and Inductive Inference. Acco, ISBN: 978 90 334 8297 7.
- [12]. Quinlan, J. R., “Induction of Decision Trees”, *Machine Learning*, 1: 81-106 (1986).
- [13]. Breiman, L., “Random Forests”, *Machine Learning*, 45(1): 5-32 (2001).
- [14]. Pal, M., “Random Forest Classifier for Remote Sensing Classification”, *International Journal of Remote Sensing*, 26(1): 217–222 (2005).
- [15]. Cover, T., and Hart, P., “Nearest Neighbor Pattern Classification”, *IEEE Transactions on Information Theory*, 13(1): 21-27 (1967).
- [16]. Soman, K. P., Loganathan, R. and Ajay, V., “Machine learning with SVM and other kernel methods”, PHI Learning Pvt. Ltd., Delhi/India, 1-10 (2009).
- [17]. Jiang, Z. G., Fu, H. G. and Li, L. J., “Support Vector Machine for Mechanical Faults Classification”, *Journal of Zhejiang University Science*, 6 (5): 433-439 (2005).
- [18]. Ben-Hur, A. And Weston, J., “A User’s Guide to Support Vector Machines”, *Methods in Molecular Biology*, 609: 223-239 (2010).
- [19]. Kavzaoğlu, T. ve Çölkesen, İ., “Destek Vektör Makineleri ile Uydu Görüntülerinin Sınıflandırılmasında Kernel Fonksiyonlarının Etkilerinin İncelenmesi”, *Harita Dergisi*, 2010(144): 73-82 (2010).
- [20]. Burges, C. J. C., “A Tutorial on Support Vector Machines for Pattern Recognition”, *Data Mining and Knowledge Discovery*, 2 (2): 121-167 (1998).
- [21]. Cortes, C. and Vapnik, V., “Support-Vector Network”, *Machine Learning*, 20(3): 273–297 (1995).
- [22]. Osuna, E. E., Freund, R., Girosi, F., “Support Vector Machines: Training and Applications”, *Massachusetts Institute of Technology and Artificial Intelligence Laboratory Report*, 8-10 (1997).

- [23]. Orhan, U. and Adem, K., “The Effects of Probability Factors in Naive Bayes Method”, Elektrik-Elektronik ve Bilgisayar Mühendisliği Sempozyumu, Bursa, 722-724 (2012).
- [24]. Krishna, P. R. and De, S. K., “Naive-Bayes Classification using Fuzzy Approach”, Third International Conference on Intelligent Sensing and Information Processing, Bangalore/India, 61-64 (2005).
- [25]. Türkoğlu, F., “Melez Yaklaşımlarla Türkçe Dokümanlarda Yazar Tanıma”, Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, 15-17 (2006).
- [26]. Harrell, F. E., “Regression Modeling Strategies: With Applications to Linear Models, Logistic Regression and Survival Analysis”, Springer-Verlag, New York, 215-267 (2001).
- [27]. Hosmer, D. W., Lemeshow, S. and Sturdivant, R. X., “Applied Logistic Regression 3rd Ed.”, Wiley&Sons Publications, Canada, 8-35 (2013).
- [28]. Washington, S. P., Karlaftis, M. G. and Mannering, F., “Statistical and Econometric Methods For Transportation Data Analysis 2nd Ed.”, Chapman and Hall/CRC, Boca Raton/FL, 263-265 (2003).
- [29]. Stock, J. H. and Watson, M. W., “Introduction to Econometrics 2nd Ed.”, Addison-Wesley, Boston, 389-390 (2007).
- [30]. Leech, N. L., Barrett, K. C. and Morgan, G. A., “SPSS For Intermediate Statistics: Use and Interpretation 2nd Ed.”, Lawrance Erlbaum Associates Publishers, New Jersey, 109-110 (2004).
- [31]. Arı, E., and Yıldız, Z., “Parallel Lines Assumption in Ordinal Logistic Regression and Analysis Approaches”, International Interdisciplinary Journal of Scientific Research, 1(3): 8-23 (2013).
- [32]. Büyüköztürk, Ş., Çokluk Bökeoğlu, Ö. ve Şekercioğlu, G., “Sosyal Bilimler İçin Çok Değişkenli İstatistik SPSS ve LISREL Uygulamaları”, Pegem Akademi Yayıncılık, Ankara, 59-65 (2010).
- [33]. Aksaraylı, M. ve Saygın, Ö., “Algılanan Hizmet Kalitesi ve Lojistik Regresyon Analizi ile Hizmet Tercihine Etkisinin Belirlenmesi”, Dokuz Eylül Üniversitesi Sosyal Bilimler Enstitüsü Dergisi, 13(1): 21-37 (2011).
- [34]. Üruk, E., “İstatistiksel Uygulamalarda Lojistik Regresyon Analizi”, Yüksek Lisans Tezi, Marmara Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, 5-6 (2007).

KURUMSAL BİLGİ SİSTEMİ UYGULAMALARI İÇİN POPÜLER PROJE YÖNETİM YAKLAŞIMLARI

Fatih Yücalar*

**Manisa Celal Bayar Üniversitesi, Hasan Ferdi Turgutlu Teknoloji Fakültesi, Yazılım Mühendisliği Bölümü,
Turgutlu, Manisa*

fatih.yucalar@cbu.edu.tr

** Sorumlu Yazar*

1. GİRİŞ

Günümüzde Endüstri 4.0 kavramının ortaya çıkmasıyla yani sanayinin dijitalleşmesi ile beraber tüm sektörleri etkileyen bir değişime tanıklık etmekteyiz. Bu değişimin yeni iş modellerinin ortaya çıkmasına ve birçok sistemin yeniden şekillenmesine neden olduğu görülmektedir. Kullanılan teknolojiler katlanarak artan bir hızla günlük olarak değil, anlık olarak değişmekte ve gelişmektedir. Bu noktada işletmelerde kullanılan kurumsal bilgi sistemlerinin günün değişen teknolojik koşullarına göre yeniden yapılandırılması veya ihtiyaçlar doğrultusunda geliştirilmesi gerekmektedir. Kurumsal bilgi sistemleri, işletme içerisinde yer alan insan gücü, malzeme, finans, zaman gibi bütün kaynakların etkin bir şekilde kullanılmasına olanak sağlayan uygulamalar olarak karşımıza çıkmaktadırlar. Ayrıca, kurumsal bilgi sistemi uygulamaları, işletme içerisindeki iş süreçlerinin otomatize edilmesini gerçekleştirmek üzere geliştirilen, karmaşık ilişkilere sahip büyük miktardaki verinin görüntülenmesini, işlenmesini ve saklanmasını sağlayan uygulamalardır.

Küreselleşen dünyada işletmeler, varlıklarını koruyarak rakipleri ile rekabet edebilmeleri için dijital dönüşüm sürecini göz önünde bulundurarak yönetim stratejilerini ve iş yapış şekillerini yeniden değerlendirmelidirler. Dijital dönüşüm, teknolojinin tüm paydaşlar tarafından kullanılmasını ve kurum varlıklarının dijitalleşmesini gerektirmektedir. Dijitalleşme ile beraber kurumlar iş yapış şekillerine göre Müşteri İlişkileri Yönetimi (Customer Relationship Management – CRM), Kurumsal Kaynak Planlaması (Enterprise Resource Planning – ERP), Tedarik Zinciri Yönetim Sistemleri (Supply Chain Management Systems – SCMS), İş Zekâsı Çözümleri (Business Intelligence Solutions – BIS), İnsan Kaynakları Yönetim Sistemleri (Human Resources Management Systems - HRMS), Kurumsal Uygulama Entegrasyonu (Enterprise Application Integration – EAI) gibi çok çeşitli yazılım çözümleri kullanmaya başlamışlardır. Kurumsal bilgi sistemi uygulamalarının geliştirilmesi noktasında bu uygulamaların işletmenin ihtiyaçlarına cevap verebilecek şekilde yapılandırılması, zamanında ve belirlenen bütçeye göre tamamlanması önemlidir. Bunu sağlamak için proje yönetim süreçlerinin doğru bir şekilde ele alınması ve takip edilmesi gerekmektedir. Etkin proje yönetiminin yanında yazılım yaşam-döngüsü adımlarının doğru takip edilmesi de önemlidir. Kullanılan yaşam-döngüsü veya çevik yazılım geliştirme metodolojisine uygun olarak her adımda planlama, dokümantasyon ve test işlemleri gerçekleştirilmelidir. Özellikle, proje ile ilgili gereksinimlerin eksiksiz ve doğru bir şekilde belirlenmesi ve analiz edilmesi projenin başarısı için önem arz etmektedir. İşletme hedeflerinin ve ihtiyaçlarının tanımlanması ile beraber paydaşlara değer katan çözümler önerilerek işletme içerisindeki değişimin başlamasına “iş analizi” öncülük eder. İş analizi yapılırken işletmenin sahip olduğu yetkinlikler göz önünde bulundurularak iş ihtiyaçları belirlenir. İş ihtiyaçları ve bu iş ihtiyaçlarına uygun çözüm stratejilerinin belirlenmesi ile beraber kurumsal bilgi sistemlerinin geliştirilmesine başlanılır.

Son yıllarda kurumsal bilgi sistemi uygulamalarının geliştirilmesi noktasında Scrum (Schwaber, 2004), XP (Beck, 2004), RUP (Kruchten, 2003), Kanban (Stellman ve Greene, 2013), Lean Practices (Miletić ve Miletić, 2017) gibi çevik yazılım geliştirme metodolojileri kullanılmaktadır. Çoğu kaynakta bir proje yönetim felsefesi olarak ele alınan bu çevik metodolojiler,

KURUMSAL BİLGİ SİSTEMİ UYGULAMALARI İÇİN POPÜLER PROJE YÖNETİM YAKLAŞIMLARI

işletmenin bilgi sisteminden ihtiyaç duyduğu gereksinimlerin belirlenen zaman ve bütçede karşılanmasını sağlamada başarılı olmuştur. Çevik metodolojilerin dışında yazılım projelerinin yönetilmesinde disiplinli yöntemler olarak ta adlandırılan ISO 12207 (Değerli ve Özbudak 2017), CMMI (Chrissis, Konrad & Shrum, 2011), ISO 15504 (Rout, 2004) gibi yazılım süreç geliştirme ve iyileştirme modelleri kullanılabilir. Bu modeller yazılım projeleri içerisindeki süreçlerin oluşturulması, yönetilmesi, değerlendirilmesi ve iyileştirilmesi konusunda birer rehber olarak yer almaktadırlar.

Yazılım projeleri için gerekli planlama, kontrol, izleme faaliyetlerini hem çevik metodolojilerin hem de disipline yöntemlerin kullanımı ile gerçekleştirilmesi mümkündür. Diğer projelerin aksine yazılım projeleri, işletmenin ihtiyaç duyduğu gereksinimleri karşılamalıdır. Proje yönetimi içerisinde bu gereksinimleri karşılamak için projeye yönelik hedefler ve kurumsal paydaşlar belirlenmelidir. Gereksinimlerin tamamı karşılandığında, proje yönetimi ile ilgili hedefler de yerine getirilmiş olacaktır. Proje yönetim süreçlerine gereken önem verilmez ise, işletme projenin mevcut durumu hakkında herhangi bir bilgiye sahip olamayacaktır. Ayrıca, projenin gelecekte istenilen hedefleri karşılayıp karşılayamayacağı konusunda herhangi bir çıkarımda bulunamayacaktır. Bu tür belirsizlikler projenin başarısızlığına davetiye çıkaracaktır.

Bu çalışmada kurumsal bilgi sistemi uygulamalarını geliştirirken etkin bir proje yönetim sürecinin nasıl oluşturulabileceği hakkında kullanılacak proje yönetim yaklaşımlarından bahsedilecektir. Çalışma içerisinde proje, proje yönetimi ve yazılım proje yönetimi gibi temel kavramlar ele alınacaktır. Yazılım proje yönetiminde popüler olarak kullanılan çevik yazılım geliştirme yöntemlerinden bahsedilecektir. Bu yöntemler içerisinde en fazla bilinen ve kullanılan XP, Scrum, RUP, Kanban ve Lean Practices yöntemlerinin temel felsefesine ve aralarındaki farklılıklara değinilecektir. Ayrıca, proje yönetimi konusunda ele alınabilecek disiplinli yöntemlerden biri olan ve yazılım projelerinde son 30 yıldır en fazla tercih edilen CMMI modeline yer verilecektir. CMMI modelinin yapısı, model gösterimleri ve CMMI'nin proje yönetimi ile ilgili süreç alanları bu başlık altında incelenecektir. CMMI modelinin kullanımı ile proje yönetiminde başarı nasıl sağlanır sorusunun cevabı da bulunmaya çalışılacaktır.

2. PROJE YÖNETİMİ

2.1. Proje Nedir?

Proje kavramı ile ilgili olarak literatürde bakıldığında pek çok tanım vardır. Proje, belirli bir amaç ve kapsam doğrultusunda başlangıç ve bitiş süresi belli olan, tanımlanmış bütçe ile bir yöntem veya metodolojinin takip edilmesine yönelik olarak gerçekleştirilen planlanmış aktiviteler bütünüdür. Başka bir tanım ile proje, benzersiz bir ürün veya hizmetin yaratılmasını sağlamaya yönelik gerçekleştirilen faaliyetler dizisi olarak ta ifade edilebilir. Proje kavramı kısaca, belirli bir sonuca ulaşmak için tamamlanması gereken bir dizi görev olarak ta tanımlanabilir. Projeler, bir işletmeye yenilik getirmek üzere bir değişim ihtiyacı sonucu ortaya çıkarlar. Herhangi bir kurumsal bilgi sisteminin daha ileri bir düzeye gelebilmesi amacıyla bu değişim ihtiyaçları yeni projeler olarak ele alınabilir. Projeler genellikle, işletme tarafından tanımlanabilen bir problemin çözümüne yöneliktir.

Günümüzün refah seviyesini oluşturan bütün endüstriyel araçlar, üretimi yapılan her türlü ürün, çevremizde gördüğümüz binalardan tutun hemen hemen her şey çeşitli projelerin ürünleri olarak karşımıza çıkmaktadır. Bu projelerin planlanması ve yönetilmesi ile insanlığa fayda sağlayacak, ihtiyaçlarını giderecek ve problemlerine çözüm önerileri getirecek ürünlerin ortaya çıkarılması sağlanmıştır.

2.1.1. Projelerin Karakteristik Özellikleri

Projeler, işletme tarafından üstlenilen günlük faaliyetlerden ziyade, özel olarak ele alınan uzun vadede işletmenin yaşam gücünü etkileyen faaliyetlerdir. Bir proje karakteristik olarak aşağıdaki özelliklere sahiptir:

KURUMSAL BİLGİ SİSTEMİ UYGULAMALARI İÇİN POPÜLER PROJE YÖNETİM YAKLAŞIMLARI

- **Zaman Çizelgesi (Timeline):** Her bir proje, başlangıç ve bitiş noktası ile ölçülebilen belirli bir zaman çizelgesine sahiptir.
- **Kaynaklar (Resources):** Her bir proje sınırlı bir sermaye ve insan gücü kaynağına sahiptir.
- **Araçlar (Tools):** Projelerin planlanması noktasında özel tip araç ve teknikler (Gantt şemaları, vb.) kullanılır.
- **Ekipler (Teams):** Projeler içerisinde çeşitli alanlarda uzmanlık isteyen ekiplere ihtiyaç vardır.

Bu verilerin dışında projelerin sahip olduğu özelliklerden biri tekrarlanmamalarıdır, yani bir kereye mahsus yapılırlar. Projelerin gerçekleştirilmesi noktasında mutlaka plan yapılması gerekir. Her proje işletmenin ihtiyaç duyduğu özel hedeflere veya ürünlere yönelik olarak belirli bir amaç doğrultusunda gerçekleştirilir. İstenilen sonuca ulaşıldığında proje sonlandırılır.

2.1.2. Proje Yaşam Döngüsü

Tipik olarak bir proje; başlatma (initiation), proje tanımlama (project definition), yapılabirlik çalışması (feasibility study), proje yürütme (project execution) ve proje sonucu (project conclusion) olmak üzere beş aşamadan oluşmaktadır. Bu aşamaların her birinin, projenin genel başarısı üzerinde kendi önem ve etkisi vardır.

Başlatma Aşaması: Projenin bu aşamasında müşterilerden alınan geri bildirimler analiz edilir. Müşteriden gelen talepleri karşılamak üzere mevcut ürünün düzenlenmesi yoksa sıfırdan yeni bir ürünün geliştirilmesi gerektiği konusunda beyin fırtınası yapılır.

Proje Tanımlama: Projenin bu aşamasında, müşteriler tarafından ortaya atılan problemin çözümünün tanımlanmasına yönelik çaba harcanılmaktadır.

Yapılabirlik Çalışması: Bu aşamada projenin planlaması yapılır ve projeye ilişkin belirli dönüm noktaları (milestones) belirlenir.

Proje Yürütme: Bu aşamada, önceki aşamada oluşturulan tüm faaliyetler ve dönüm noktaları zamanında ve düzenli bir şekilde yürütülür. Ayrıca, bu aşamada maksimum seviyede tüm kaynaklardan yararlanılır.

Proje Sonucu: Bu, projenin son aşamasıdır. Bu aşamada, nihai ürün veya hizmet ticari üretim için operasyon ekibine teslim edilir.

2.2. Proje Yönetimi

Proje yönetimi, bir projeyi belirli gereksinimlere göre tamamlamak için bilgi, beceri, araç ve teknikleri uygulama pratiğidir. Başka bir ifade ile proje yönetimi; proje hedeflerine ulaşmak için mevcut kaynakları en etkin biçimde planlama ve proje aktivitelerini kontrol etme süreci olarak tanımlanabilir (Tapsız, 2018). Proje yönetimi kapsamında; problemi tanımlama, problemi çözmek için bir plan oluşturma ve daha sonra problem çözümlene kadar bu plan üzerinden ilgili yürütme işlemini sağlama faaliyetleri gerçekleştirilir. Kâğıt üzerinde bu kulağa basit gelebilir, ancak sürecin her aşamasında projeye dahil olan birçok etken vardır (Project Management Basics, 2020).

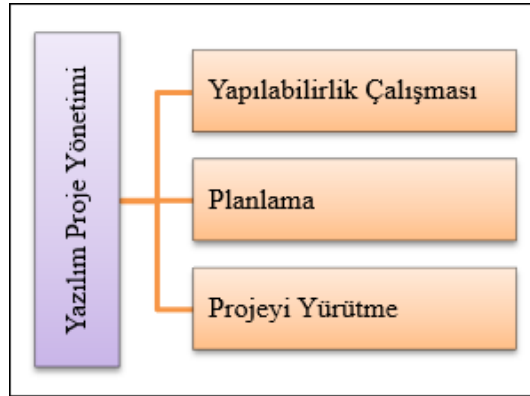
Etkin bir proje yönetim sürecinin takip edilmesi sonucu, bir proje belirlenen bütçe dahilinde zamanında tamamlanmışsa ve müşterinin ihtiyaç duyduğu performans gereklerini eksiksiz bir şekilde yerine getirmişse başarılı sayılır. Son yıllarda geliştirilen projelerin teknolojik gelişmelerin etkisi ile daha kapsamlı bir içeriğe sahip olması, küreselleşmenin getirdiği rekabet ortamı, uluslararası iş yapabilme olanaklarındaki artış, bilgi sistemleri üzerinden işletme faaliyetlerinin düzenli olarak izlenebilmesi ve gerekirse müdahale edilebilmesi gibi nedenlerden dolayı proje yönetiminin popülaritesi artmıştır.

2.3. Yazılım Proje Yönetimi

Yazılım proje yönetimi, yazılım ve web projelerinin planlanması, programlanması, kaynak tahsisi, yürütülmesi, takibi ve teslimi için ayrılmış olan proje yönetimi dalını ifade eder.

Yazılım mühendisliği açısından proje yönetimi, geleneksel proje yönetiminden farklıdır. Bunun nedeni ise yazılım projelerinin birden fazla test, güncelleme ve müşteri geri bildirim gerektiren benzersiz bir yaşam döngü sürecine sahip olmasıdır. Son yıllarda, bilgi teknolojileri ile ilgili projelerin çoğu, müşteri ve paydaş geri bildirimlerine bağlı olarak artan iş hızına ayak uydurmak için çevik tarzda yönetilir. Bunun nedeni müşterilere talep edilen ürünü eksiksiz bir şekilde kısa süre teslim ederek, müşteri memnuniyetini arttırmaktır.

Bir kurumsal bilgi sisteminin geliştirilmesine yönelik olarak yazılım proje yönetimi üç temel adımı içermektedir. Şekil 1'de yazılım proje yönetiminin bu üç temel adımı görülmektedir.



Şekil 1: Yazılım proje yönetimi adımları

Yazılım proje yönetimi içerisindeki ilk temel adım yapılabilirlik çalışmasıdır. Herhangi bir proje yöneticisi binlerce (veya milyonlarca) dolara mal olabilecek bir yazılım projesine yeşil ışık yakmadan önce, yapılabilirlik çalışmasını görmek ister. Bu çalışma, yazılım projesine başlamadan önce yapılan ön değerlendirme çalışmasıdır. Yazılım projesi ile ilgili olarak işlevsel ve işlevsel olmayan gereksinimler toplanılır (Schach, 2010). Elde edilen bu gereksinimlerin proje ile ilgili gerçekleştirilmek istenen hedefleri karşılayıp karşılamadığına bakılır. Ayrıca, maliyet-fayda analizi yapılarak yazılım geliştirme ve işletme maliyetleri ile yeni sistemin getirileri hesaplanır (Hughes & Cotterel, 2009). Sonuç olarak yapılabilirlik çalışması, yazılım geliştirme gereklilik ve önceliklerinin belirlendiği bir stratejik planlama gibidir.

İlk adımda gerçekleştirilen yapılabilirlik çalışması yazılım projesinin yapılabilir olduğunu gösteriyorsa ikinci adım olan proje planlama çalışmalarına başlanılır. Büyük yazılım projelerinin geliştirilmesi noktasında başlangıçta projenin tamamı için detaylı planlama yapılmaz. Başlangıç aşaması için detaylı bir plan, projenin tamamı için ise bir taslak plan yapılır. Çünkü, yazılım projesinin önceki aşamaları tamamlandıktan sonra daha doğru ve detaylı bilgilere sahip olunacağından, sonraki aşamaların planlanması başlangıç adımına bırakılır.

Planlama aşamasının tamamlanmasının ardından projenin yürütülmesi adımına geçilir. Projenin yürütülmesi adımı yazılım projesinin mimarisine uygun olarak ele alınan tasarım, kodlama, entegrasyon ve test alt-safhalarını içerir (Hughes & Cotterel, 2009). Projenin yürütülmesi, proje yönetim planında belirtilen yazılım yaşam-döngüsü, çevik yazılım geliştirme metodolojisi veya bir disipline model üzerinden takip edilebilir.

2.3.1. Yazılım Projelerini Diğer Projelerden Ayıran Farklar

Genel proje yönetimi içerisinde yer alan birçok teknik, aynı zamanda yazılım proje yönetimi için de uygulanmaktadır. Ancak, yazılım projeleri diğer projelerden farklı bir yapıya sahiptir. Bu konuda Fred Brooks tarafından yapılan bir çalışmada (Brooks, 1987), yazılım projelerini diğer

KURUMSAL BİLGİ SİSTEMİ UYGULAMALARI İÇİN POPÜLER PROJE YÖNETİM YAKLAŞIMLARI

projelerden farklılaştıran karakteristik özellikleri; görünmezlik, karmaşıklık, uygunluk ve esneklik olarak tanımlamıştır.

Brooks tarafından yapılan bu çalışmada, görünmezlik özelliği; köprü gibi fiziksel bir yapıyı inşa ederken bu köprünün gelişiminin gözlemlenebileceği, ancak bir yazılım projesi üzerindeki gelişimin ise hemen gözlemlenemeyeceği ifade edilmiştir. Ayrıca, yazılım proje yönetiminin, görünmezliği görünür hale getirme süreci olarak ele alınabileceği belirtilmiştir. Bu çalışmada karmaşıklık özelliği ise, yazılım projeleri için yapılan harcamalar dikkate alındığında, diğer projelere göre daha karmaşık bir yapıya sahip olduğu ifade edilmiştir. İlgili çalışmada uygunluk özelliği için ise, geleneksel mühendislerin fiziksel sistemleri belirli sabit kurallar çerçevesinde belli malzemeler kullanılarak gerçekleştirdiği, ancak yazılım geliştiricilerin yazılım projelerini müşteri gereksinimlerine uygun şekilde gerçekleştirmek zorunda olduğuna değinilmiştir. Esneklik özelliği için ise, güç olarak görünen bir yazılımın kolayca değiştirilebilecek esnek bir yapıya sahip olduğu ifade edilmiştir. Bu noktada, yazılım projeleri planlanırken ileride olabilecek olası değişiklikler göz önünde bulundurularak yazılım mimarisinin esnek bir yapıya sahip olmasına dikkat edilmelidir (Hughes & Cotterel, 2009).

3. PROJE YÖNETİM YAKLAŞIMLARI

Her proje yöneticisi, doğru proje yönetim yaklaşımını seçmenin işi doğru yapmak kadar önemli olduğunu bilir. Sektörde kullanılan Scrum, XP (Extreme Programming – Uç Programlama), Kanban, RUP (Rational Unified Process), Lean Practices (Yalın Uygulamalar) gibi birçok çevik proje yönetim yaklaşımı bulunmaktadır. Bunların dışında disiplinli yöntemler içerisinde yer alan ve sektörde kabul görmüş bir süreç modeli olan CMMI yer almaktadır. Bu proje yönetim yaklaşımlarının her birinin kendine has kuralları, ilkeleri, süreçleri ve uygulamaları vardır.

Hangi proje yönetim yaklaşımının seçilmesi ve uygulanması gerektiği tamamen üstlenilecek projenin türüne bağlıdır. Proje yöneticileri, kaynakların ve zamanın kullanımını en üst düzeye çıkarmak amacıyla bir proje yönetim yaklaşımını tercih etmek zorundadırlar.

Akılda tutulması gereken bir şey, aralarından seçim yapabileceğiniz birçok proje yönetim yaklaşımı olsa da “doğru” proje yönetim yaklaşım diye bir şey olmadığıdır. Bunun anlamı, her proje için mükemmel olan tek bir proje yönetim yaklaşımı yoktur. Projelerin kapsamı ve gereksinimleri birbirinden farklıdır, yani uygulanması gereken doğru proje yönetim yaklaşımı da değişecektir.

Bu başlık altında kurumsal bilgi sistemi uygulamalarının geliştirilmesinde kullanılacak popüler proje yönetim yaklaşımlarından bazılarını değinilecektir.

3.1. Çevik Yöntemler

Yazılım endüstrisinde sıkça kullanılan proje yönetim yaklaşımlarından biri “Çevik (Agile)” yöntemlerdir. Bu yöntemler, yinelenmeli ve artımlı yazılım projelerinde kullanılmak üzere oldukça idealdir. Çevik yaklaşım, müşteriler ile kendi kendini organize eden yazılım ekiplerinin ortak çabalarıyla taleplerin ve çözümlerin geliştiği bir süreçtir. Başlangıçta yazılım geliştirme için yaratılan bu süreç, yazılım endüstrisinin son derece rekabetçi ve sürekli değişen taleplerini karşılamayan Şelale (Waterfall) yönteminin yetersizliklerine çözüm olarak ortaya atılmıştır. Ayrıca, müşterilerin talep ettiği yazılım ürünlerine ilişkin sürümlerin belirlenen sürede çıkarılamaması, müşterilerden gelen değişiklik taleplerine geri dönüş sağlanamaması, geliştirilen yazılım ürünleri içerisindeki hataların geç fark edilmesi, yazılım geliştirme süreci içerisinde gelen yeni taleplere göre yazılımın kendi yapısını geliştirememesi gibi çeşitli sorunlar ile karşılaşılması çevik yöntemlerin ortaya çıkışını tetiklemiştir (Çamoğlu, Akbayır, Yücalar & Bayraklı, 2010).

2001 yılında Amerika'nın Utah eyaletinde bir araya gelen 17 endüstri liderinin ortak görüşü ile “Çevik Yazılım Geliştirme Manifestosu” olarak adlandırılan bir bildiri hazırlanmış ve yayınlanmıştır. Yayınlanan bu “Çevik Yazılım Geliştirme Manifestosu”, 4 temel değer (Manifesto for Agile Software Development, 2020) ve 12 temel ilkedden (Principles Behind the Agile Manifesto, 2020) oluşmaktadır. Çevik proje yönetim yaklaşımları bu temel değer ve ilkeler üzerine kurulmuştur.

KURUMSAL BİLGİ SİSTEMİ UYGULAMALARI İÇİN POPÜLER PROJE YÖNETİM YAKLAŞIMLARI

Çevik yazılım geliştirme manifestosunun amacı; yinelemeli gelişimi, ekip işbirliğini ve değişikliği kabul etmeyi destekleyen, açık ve ölçülebilir bir yapı sağlayarak yazılım geliştiriminin daha iyi yollarını ortaya çıkarmaktır.

Çevik yaklaşımlar uyarlanabilir bir yapıya sahip olduğu için esneklik gerektiren, karmaşıklık veya belirsizlik düzeyi yüksek olan projelerin yönetiminde yaygın olarak kullanılırlar (Yücalar & Borandağ, 2018). Bu yaklaşımlar ürünü oluşturmak ve ürün gelişimini izlemek için; ürün vizyon beyanı, ürün yol haritası, ürün gereksinim listesi (product backlog), sürüm planı, önceliklendirilmiş gereksinim listesi (sprint backlog) ve artım (increment) olmak üzere altı temel çıktıyı kullanırlar. Bu özellikleri ile çevik yöntemler, sağladığı esneklik ve işbirliği ile sürekli iyileştirmeye ve yüksek kaliteli sonuçlara önem veren bir proje yönetim yaklaşımı olarak kendini kanıtlamaktadır (Pichler, 2010).

Şekil 2’de görüldüğü üzere çevik yazılım geliştirme yöntemleri kendi içerisinde aynı felsefeyi sürdüren ancak uygulama aşamasında pratikleri farklılaşan çeşitli yöntemlere ayrılmaktadırlar (Highsmith, 2002).



Şekil 2: Çevik Yazılım Geliştirme Yöntemleri

3.1.1. Scrum

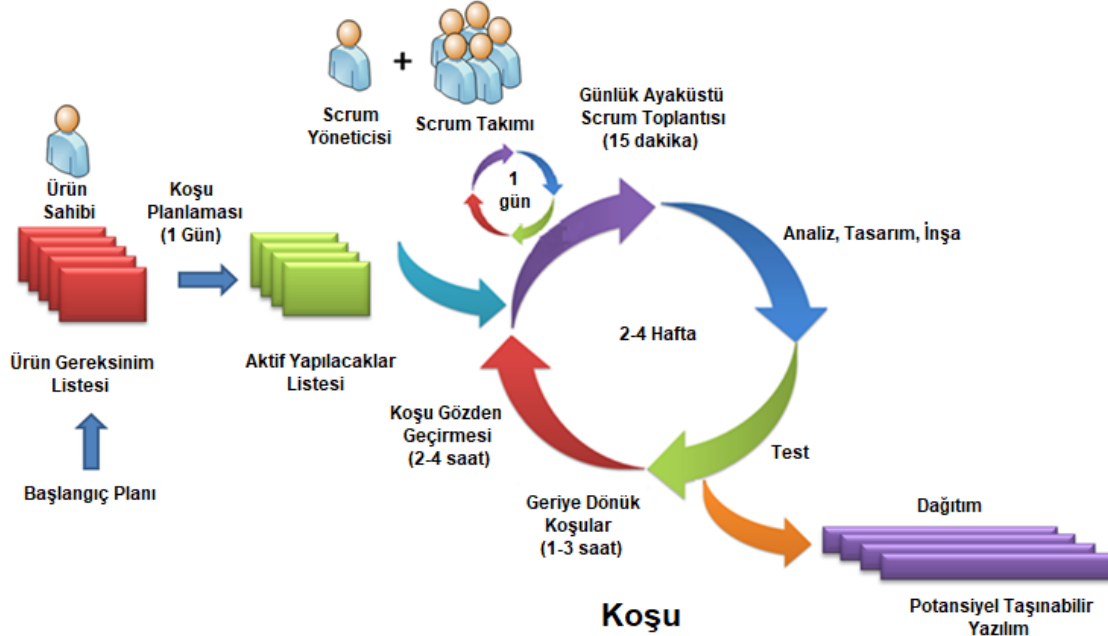
Scrum, müşterilere yapılacak olan ürün teslimatlarını iyileştirmek için süreç ve ilkeler öneren bir proje yönetim yaklaşımıdır. Jeff Sutherland ve Ken Schwaber tarafından 1990’ların ortalarında geliştirilen bu proje yönetim yaklaşımı, yazılım projelerinde çevik prensiplerini uygulama noktasında basit bir çerçeve sunar. Scrum, en popüler çevik yazılım geliştirme yaklaşımlarından biridir. Scrum’ın amacı, projelerde iletişimi, ekip çalışmasını ve geliştirme hızını arttırmaktır. Scrum taahhüt (commitment), cesaret (courage), odaklanma (focus), açıklık (openness) ve saygı (respect) olmak üzere beş değerden oluşmaktadır.

Scrum, sorumluluk, işbirliği ve yinelemeli bir gelişim yolu ile karmaşık yazılım ürünlerini küçük birimlere (sprint) bölerek geliştirmeyi ve teslimi öngörür. Scrum’ı diğer çevik proje yönetim yaklaşımlarından ayıran şey belirli rolleri, olayları ve bileşenleri kullanarak nasıl çalıştığıdır.

Scrum’ın çalışma mantığına bakıldığında, bir ürün sahibi (product owner) tarafından tanımlanmış (product backlog) ve önceliklendirilmiş (sprint backlog) gereksinim listelerinde yer alan öğeler (items) dikkate alınarak en fazla 9 kişiden oluşan küçük bir ekip proje üzerinde çalışır. Şekil 3’te görüldüğü üzere çalışma, genellikle 2-4 haftalık bir gelişim döngüsü olan sprintler’e bölünür. Bu süreçte ekibin ilerleme ve engelleri rapor ettiği 15 dakikalık ayaküstü “Günlük Scrum Toplantıları” yapılır. Her bir sprint sonunda ise “Sprint Gözden Geçirme Toplantıları” gerçekleştirilir. Bu toplantı, ilgili sprint’in Ürün Sahibi Bitti Tanımı’ndan geçip geçmediğini kontrol etmek amacıyla yapılır.

KURUMSAL BİLGİ SİSTEMİ UYGULAMALARI İÇİN POPÜLER PROJE YÖNETİM YAKLAŞIMLARI

Ekibe, bir Scrum Yöneticisi (Scrum Master) liderlik eder. Bu yönetici, ekibi Scrum'a adapte ederek, ekibin Scrum'ın temel değerlerine, ilkelerine ve pratiklerine bağlı kalmasını sağlar (Cohn, 2010). Ayrıca, ekibin dış etkenlerden korunması, ekip üyelerinin sahip olduğu yeteneklerin geliştirilmesi ve iyileştirilmesi de bu yöneticinin sorumluluğundadır (Çamoğlu, Akbayır, Yücalar & Bayraklı, 2010).



Şekil 3: Scrum'ın çalışma mantığı (Yücalar & Borandağ, 2018)

Şekil 3'te görüldüğü üzere Scrum Yöneticisi, her bir sprint içerisinde günlük olarak o gün ne yapılacağı ve bir önceki gün ne yapıldığı sorularına günlük 15 dakikalık ayaküstü toplantılar yaparak cevap aramaktadır.

3.1.2. Uç Programlama

Uç programlama (eXtreme Programming – XP), 1999'da Kent Beck tarafından ortaya atılan bir proje yönetim metodolojisidir (Beck, 2004). Bu proje yönetim metodolojisi, değişen müşteri gereksinimlerini karşılamak ve yazılım kalitesini artırmak için gerekli değerleri ve süreçleri tanımlamaktadır. Basitlik, iletişim, geri bildirim ve cesaret olmak üzere dört temel değere sahip olan XP, yapı olarak Scrum'a çok benzerdir. XP, yazılım projelerinde kolaylığı ve esnekliği sağlamak üzere 12 farklı pratiği öngörmektedir (Borandağ, Yücalar, Erdoğan, & İnce, 2009). Bu pratikler; planlama oyunu, ekipte müşteri, önce test, basit tasarım, çiftli programlama, sürekli entegrasyon, kısa aralıklı sürümler, yeniden yapılandırma, ortak kod sahiplenme, metafor (benzetim), kodlama standardı, haftada kırk saat çalışmadır.

XP'nin Scrum'dan kesin olarak ayrıldığı nokta, pratikler içindeki tanımlı kurallar veya kuralcı süreçlerdir. Bunların bazıları Scrum'a benzerdir. Ancak, yazılım geliştirme projeleri için XP'yi özel kılan kodlama ve test tasarımı konusunda teknik uygulamalarla ilgili kuralları vardır. Kullanıcı hikayeleri, test öncelikli geliştirme, çiftli programlama ve sürekli entegrasyonu içeren kurallar zorunlu olarak ele alınmaktadır.

Yüksek düzeyde müşteri memnuniyetini sağlamasından dolayı XP başarılıdır. Bunun nedeni ise müşterinin isteyebileceği her şeyi gelecekteki bir tarihte teslim etmek yerine, XP kısa zaman aralığında talep edilen yazılımın sadece müşterinin ihtiyaç duyduğu kadarının teslim edilmesini öngörür. XP, proje tamamlanmak üzereyken bile değişen müşteri gereksinimlerinin karşılanması noktasında yazılım geliştiricilere esneklik sağlar.

KURUMSAL BİLGİ SİSTEMİ UYGULAMALARI İÇİN POPÜLER PROJE YÖNETİM YAKLAŞIMLARI

XP, ekip çalışmasını desteklemektedir. Yöneticiler, geliştiriciler ve müşteriler işbirliği içerisinde, bir grup kimliği altında ekip olarak hareket ederler. XP, ekiplerin son derece üretken olmalarını sağlayan basit ama etkili bir ortam sağlamaktadır. Ekip, sorunu mümkün olduğunca verimli bir şekilde çözmek için kendi etrafında organize olur (Extreme Programming: A gentle introduction, 2020).

3.1.3. Kanban

Kanban, verimliliği artırmak için katı bir sürece ve Yalın (Lean) ilkelere odaklanan bir proje yönetim metodolojisidir (Hiranabe, 2008). Scrum'a birçok açıdan benzeyen Kanban, işbirlikçi ve kendi kendini yöneten ekiplerle erken sürümlere odaklanarak müşteri memnuniyetini arttırmayı hedefleyen bir başka popüler çevik yöntemdir. Scrum ile karşılaştırıldığında, Kanban daha az kuralcı bir yaklaşıma sahiptir. Bundan dolayı, Kanban çevik dünyasına daha yumuşak bir iniş sağlar ve daha evrimsel bir değişimi öngörür.

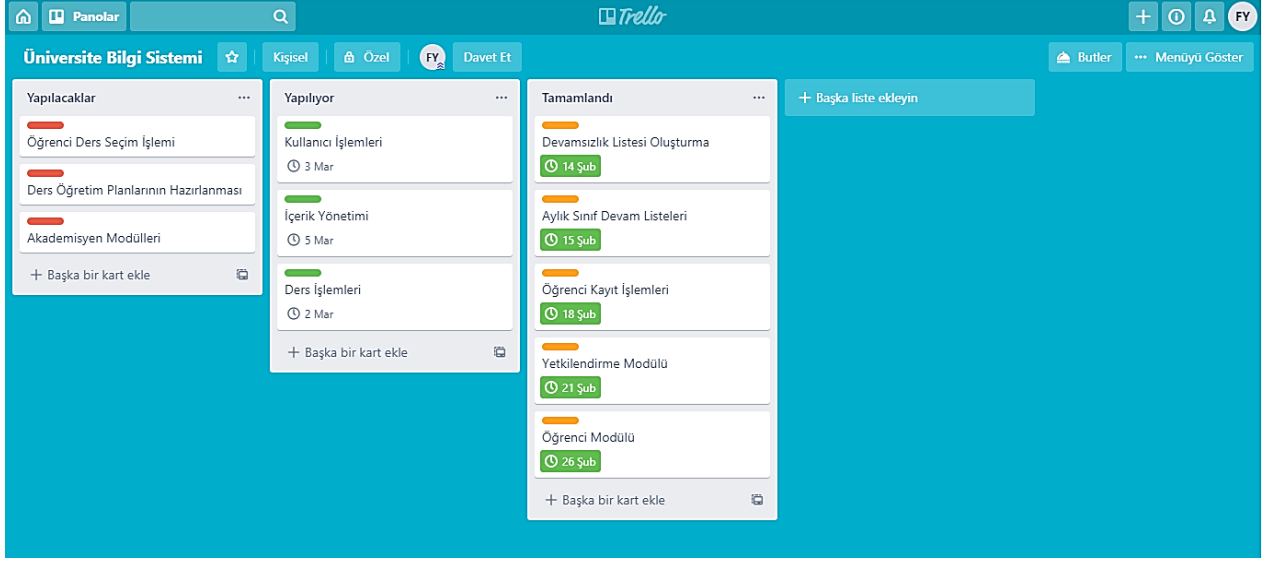
Görsel bir metodoloji olan Kanban, iş akış sürecinin resmini çizerek yüksek kalitede sonuçlar sunmayı amaçlar. Bu sayede, süreç içerisindeki darboğazlar geliştirme sürecinin erken aşamalarında tespit edilebilir. Kanban metodolojisi iş akış sürecini aydınlatır, esnektir, kurallar ile belirlenmiş roller yoktur. Ekibin gerçekten önemli olan konulara odaklanmasını sağlayarak, verimi artırmaya çalışır.

Kanban; iş akışını görselleştirme, devam etmekte olan işi sınırlama, akış yönetimi, teslim süresini ölçme, süreç politikalarını açık hale getirme ve geri bildirim döngülerini kullanma olmak üzere altı temel pratik üzerine çalışır (Raut, Wakode & Talmale, 2015). Kanban, geliştirme sürecinin çeşitli aşamalarını gösteren görsel işaretler kullanarak verimlilik sağlar. Sürece dahil olan görsel işaretler; Kanban görev tahtası (task board), Kanban kartları (cards) ve Kanban kulvarları (swimlanes)'dir .

- **Kanban görev tahtası:** Geliştirme sürecini görselleştirmek için kullanılır. Bu görev tahtası fiziksel (beyaz tahta, yapışkan notlar ve işaretleyiciler) veya dijital (Trello gibi bir çevrimiçi proje yönetim aracı) olabilir.
- **Kanban kartları:** Her Kanban kartı, iş süreci içerisinde yer alan bir iş ögesini/görevini tasvir eder. Bu kartlar ekip ile iletişim kurmak için kullanılır. İşin durumu, dönüş süresi ve yaklaşan son tarihler gibi bilgileri temsil eder.
- **Kanban kulvarları:** Yatay olarak akan Kanban kulvarları, görev tahtasındaki görevleri/öğeleri kategorilere ayırarak, bu görevlerin/öğelerin daha fazla ayırt edilmesini sağlayan görsel bir öğedir. Kanban kulvarlarının amacı, iş akışına daha iyi bir genel bakış sunmaktır.

Bir kurumsal bilgi sistemi projesi geliştirirken Kanban görev tahtası kullanılarak; projenin tamamlanmasına kadar olan süreçteki gelişim aşamaları kolaylıkla takip edilebilir. Bu takibi gerçekleştirmek için görev tahtası üzerinde "Yapılacaklar (To-Do), Yapılıyorlar (Doing) ve Tamamlandı (Done)" olarak etiketlenmiş üç sütun yer alır. Trello çevrimiçi proje yönetim aracı içinde yer alan Kanban görev panosu (Siderova, 2019) Şekil 4'te görülmektedir.

KURUMSAL BİLGİ SİSTEMİ UYGULAMALARI İÇİN POPÜLER PROJE YÖNETİM YAKLAŞIMLARI



Şekil 4: Trello çevrimiçi proje yönetim aracı içinde yer alan Kanban görev tahtası

Kanban işlerin hızlı ve kaliteli bir biçimde piyasaya sürülmesine odaklanır. Önceliklerin sık sık değiştirilebileceği operasyonel veya bakım ortamlarında kullanılmak üzere mükemmel bir yaklaşımdır. Ayrıca, Kanban yaklaşımı ürünün teslim süresini ölçerek, ürünün ne kadar sürede müşteriye teslim edilmesi gerektiğine odaklanır. Sonuç olarak, diğer çevik yöntemler gibi Kanban da yazılım sektöründe iz bırakmıştır. Sağlamış olduğu esneklikten dolayı yazılım haricindeki diğer endüstrilerde popüler olarak kullanılmaktadır.

3.1.4. Lean

Lean (yalın) metodoloji, verimlilik temasına odaklanmış yani israfı en aza indirerek müşteri değerini en üst düzeye çıkarmayı teşvik eden bir proje yönetimi metodolojisidir (Aston, 2019). Daha az kaynak kullanarak müşteri için daha fazla değer yaratmayı amaçlamaktadır. Başka bir ifade ile Lean, daha azla daha fazlasını yapmayı hedefleyen bir yaklaşımdır. Japon üretim endüstrisinden gelen Lean değerleri, israf ortadan kaldırıldıkça, üretim süresi ve maliyet azalırken kalitenin arttığını varsayar (Aston, 2019). Bu amaçla, Lean değeri belirleyerek başlar, daha sonra sürekli iyileştirme yoluyla değer akışını optimize eder ve israfı ortadan kaldırarak değeri en üst düzeye çıkarır. Şekil 5'te Lean yaklaşımının temel prensipleri görülmektedir.



Şekil 5: Lean yaklaşımının temel prensipleri

KURUMSAL BİLGİ SİSTEMİ UYGULAMALARI İÇİN POPÜLER PROJE YÖNETİM YAKLAŞIMLARI

Lean yaklaşımı; 3Ms (muda, mura ve muri) olarak da bilinen üç tür israf tanımlar:

- **Muda**, müşteriye değer katmayan herhangi bir faaliyet veya süreç ile ilgili israfın ortadan kaldırılmasıdır (Aston, 2019). Dijital dünyada, bu durum gözden geçirme kontrollerinin kaldırılması anlamına gelir.
- **Mura**, standart sürece yönelik değişimlerin yarattığı ek yükün ortadan kaldırılmasıdır (Aston, 2019). Onay süreçlerinin ve talimatların standartlaşması anlamına gelir.
- **Muri**, aşırı iş yükünün ortadan kaldırılmasıdır. Optimum çalışma kapasitesi %60-70 oranında olmalıdır (Aston, 2019). Bundan daha fazlası ve her şey yavaşlar. Yürütmeye çalışılan proje sayısının azaltılması gerekir.

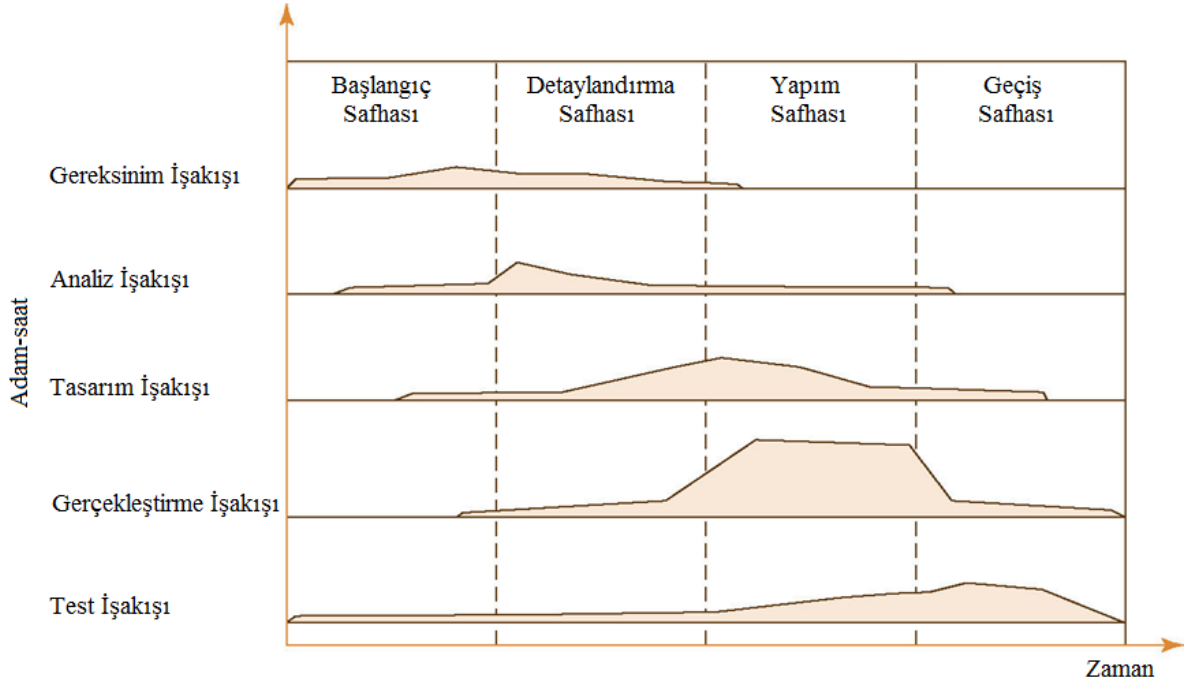
Lean felsefesi, işletmelerin değer sunabilmesi için çalışma şekillerini değiştirmesi gerektiğini ifade eder. Genellikle üretim endüstrilerinde uzmanlaşmakla karıştırılan Lean metodolojisi, aslında bir süreç aramayan, ancak iş yapma şeklini dönüştürmekle ilgilenen herhangi bir işletme veya kuruluş için idealdir.

3.1.5. Rasyonel Birleşik Süreç

Rasyonel Birleşik Süreç (Rational Unified Process – RUP), internet erişimli aranabilir bir bilgi tabanı üzerinden sunulan ve geliştirme maliyetlerini azaltmak için kullanılan bir yazılım mühendisliği sürecidir (Kruchten, 2003). Bu süreç, ekip verimliliğini artırır ve tüm kritik yazılım yaşam-döngüsü etkinlikleri için yönergeler, örnek şablonlar ve yol gösterici araçlar üzerinden en iyi yazılım uygulamalarını sunar. RUP yaklaşımını kullanan organizasyonlarda ekip üyelerinin sorumlulukları detaylıca bellidir. Burada amaç ekibin üretkenliğini arttırmaktır. Şekil 6'da görüldüğü üzere RUP; başlangıç (inception), detaylandırma (elaboration), yapım (construction) ve geçiş (transition) olmak üzere dört safhadan oluşur (Kruchten, 2003):

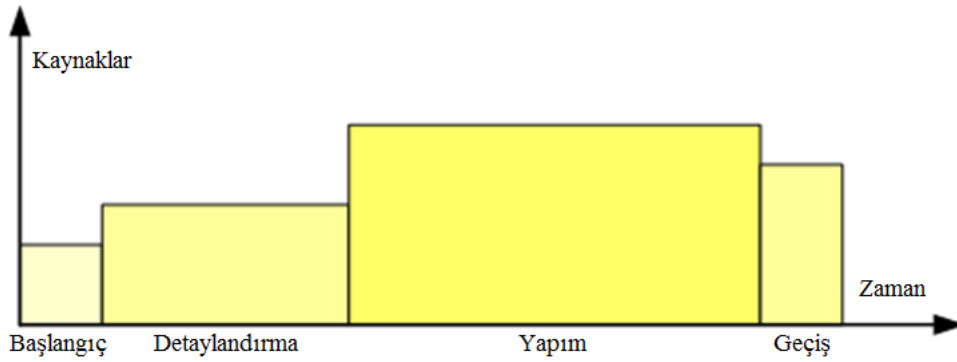
- **Başlangıç Safhası:** Proje kapsamının ve iş durumlarının (business case) belirlendiği ve tanımlandığı safhadır. İş durumu genellikle uygulama sahası yani iş hakkında belirli ayrıntıları sağlar.
- **Detaylandırma Safhası:** Geliştirilecek sisteme ilişkin başlangıç gereksinimlerinin detaylandırıldığı ve mimarinin iyileştirildiği safhadır. Ayrıca, proje kapsamında ortaya çıkması muhtemel riskleri gözlemleyip önceliklerini belirlemek ve yazılım proje yönetim planını ortaya çıkarma işlemi de bu safhada gerçekleştirilir.
- **Yapım Safhası:** Bu aşamada uygulamanın tasarımı ve kaynak kodu geliştirilir. Yazılım ürününün operasyonel – kaliteli ilk versiyonu bu aşamada üretilir.
- **Geçiş Safhası:** Müşteri gereksinimlerinin gerçekten karşılanıp karşılanmadığının tespiti geçiş safhasında gerçekleştirilir. Ayrıca, yazılım ürünü içindeki hatalar düzeltilir.

KURUMSAL BİLGİ SİSTEMİ UYGULAMALARI İÇİN POPÜLER PROJE YÖNETİM YAKLAŞIMLARI



Şekil 6: RUP'un dört temel safhasında gerçekleşen süreç (Schach, 2010)

Her safha içerisinde yeri geldiğinde azda olsa kod yazılabileceğinden, safhanın sonunda değerlendirilmek üzere çalışan bir sürüm elde edilir. Genel anlamda RUP metodolojisi kullanılarak geliştirilen projelerde, Şekil 7'de görüldüğü üzere proje takviminin %10'u başlangıç safhasına, %30'u detaylandırma safhasına, %50'si yapım safhasına ve geri kalan %10'unu ise geçiş safhasına ayırılır.



Şekil 7: RUP'un dört temel safhası için ayrılan proje takvimi

Her safhanın sonunda gerçekleştirilen iş listesi, proje yönetim planı, yazılım mimari planı, güncel risk listesi, kullanıcı el kitapları gibi çıktılar elde edilir. Bu açıdan bakıldığında kurumsal bilgi sistemleri gibi büyük ölçekli ve değişim gerektiren projelerde RUP metodolojisinin kullanımı fayda sağlayacaktır.

3.2. Disiplinli Yöntemler

Yazılım projelerinin büyümesiyle 1960'lı yıllarda ortaya çıkan yazılımda kalite sorunu 1970'lerde "yazılım krizi" olarak literatüre giren bir düzeye ulaşmıştır. 1970'li yıllarda ABD'de ki kamu kurum ve kuruluşları tarafından sipariş yolu ile satın alınan yazılımların ancak %5'i kabul edilip kullanılabilmiş, %40'ı iptal edilmiş, %55'i ise bütçe aşımı, zamanında teslim edilememesi, istenilen performansı sağlamaması veya bazı işlevsel özelliklerinden taviz verilerek kullanılabilmiştir (Erdoğan, 2009).

KURUMSAL BİLGİ SİSTEMİ UYGULAMALARI İÇİN POPÜLER PROJE YÖNETİM YAKLAŞIMLARI

1980'li yıllarda itibaren yapılan çalışmalarla, yazılımda kalite sertifikasyonu sağlama, süreç iyileştirme ve yetenek belirleme çalışmaları ile ilgili olarak çeşitli modeller ortaya konulmuştur. Son kırk yıl içerisinde disiplinli yöntemler olarak ta andlandırılan birçok model geliştirilmiştir. Bu modeller içerisinde en çok bilinen ve kullanılanları arasında ISO/IEC 12207, ISO/IEC TR 15504 (SPICE) ve CMMI sayılabilir. Bu modellere benzer birçok model daha vardır. Ancak bu çalışmada o modeller değerlendirme dışı tutulmuştur.

ISO/IEC 12207, yazılım faaliyetleri ile ilgili her türlü geliştirme, satın alma, işletme, bakım vb. alanlarda kullanılan bir süreç standardıdır. Yazılım süreçlerini tanımlar, süreç değerlendirmesine karışmaz. Geliştirme yöntemlerinden ziyade her bir alana ilişkin süreçler düzeyinde tanımlanmıştır. Süreç ile ilgili olarak çalışan ekip, süreç içerisinde neyi, nasıl yapacaklarını bilmekle kalmayıp, bu süreci ne amaçla gerçekleştirdiklerini de bilerek ortak bir hedefe doğru çalışmalarını yönlendirirler (Yücalar, 2006).

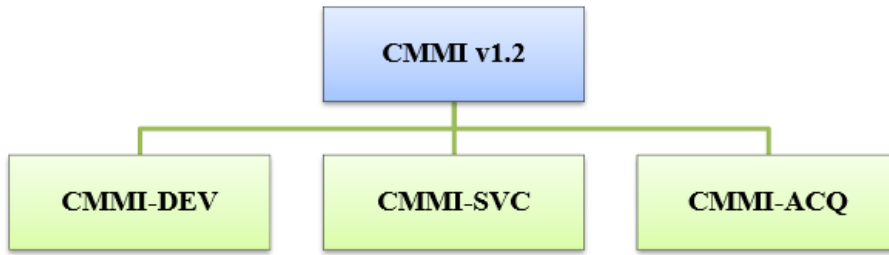
Süreçler ve yetenek düzeyleri olmak üzere iki boyutlu bir model olan ISO/IEC TR 15504, süreç iyileştirme ve yetenek belirleme amacını taşır. Yazılım mühendisliği aktiviteleri için, süreç ve süreç yeteneğini kapsayan bir referans modeldir. Bu model küçük, orta ve büyük ölçekteki her türlü projeye, organizasyona ve endüstriye kolaylıkla uyarlanabilir. Süreç içerisindeki iyileşmeyi, gelişmeyi ölçebilir. ISO 15504, herhangi bir belgelendirmeye yönelik sertifikasyon amacı taşımayan bir süreç standardıdır. Yazılım projelerinde geliştirme, satın alma, tedarik, işletim, bakım ve destek gibi süreçleri kapsar (Yücalar, 2006).

CMMI, son yıllarda yazılım yöneticilerinin yazılım süreçlerindeki performans, maliyet ve zaman tahmini gibi konularda yaşadığı sıkıntıları gidermek ve yazılımda kalite sertifikasyonunu sağlamak amacıyla geliştirilmiş bir süreç iyileştirme ve yetenek belirleme modelidir (Yücalar, Şahinaslan, Borandağ & Şahinaslan, 2010).

3.2.1. CMMI

Yazılım endüstrisinde yaşanan kötü tecrübeler sonrası yazılım projesi geliştirmek veya satın almak isteyen kurum ve kuruluşlar kendilerini güvence altına almak istemeye başladılar. Yazılım projelerinin başarısız olmasını önlemek amacıyla çeşitli arayışlar içine girdiler. Bu arayışlar içerisinde belki de en etkili olanı Amerikan Savunma Bakanlığı'nın (Department of Defense – DoD) Carnegie Mellon Üniversite'nden yardım istemesi üzerine, bu üniversite bünyesinde kurulan SEI (Software Engineering Institute) tarafından geliştirilen CMM olmuştur. CMM, etkin bir yazılım geliştirme sürecinin anahtar elemanlarını tanımlayan bir çerçeve model olarak 1991'de v1.0 olarak yayınlandı. Başlangıçta yazılım amaçlı olarak geliştirilmiş ve SW-CMM (Software CMM) adını almıştır. Özellikle yazılım alanında oldukça başarılı olmuş ve yazılım dünyasının dikkatini çekerek belki de ilk olmanın verdiği ivme ile en çok kullanılan ve bilinen model olmuştur. Daha sonra organizasyon içerisinde farklı alanlarda kullanılmak üzere, Sistem Mühendisliği Yetenek Olgunluk Modeli (SE-CMM), Entegre Ürün Geliştirme Yetenek Olgunluk Modeli (IPD-CMM), Yazım Satın Alma (SA-CMM) ve People CMM (P-CMM) gibi türevleri ortaya çıkmıştır. Sözü edilen bu modellerin kullanımı organizasyon içinde faydalar sağlamış, ancak modellerin birbirlerinden farklı ve entegre olmamış olmaları uygulama aşamasında çeşitli sorunlarla karşılaşılmasına neden olmuştur. Modeller arası birbirine karışmalar, örtüşmeler ve tezatlar oluşmuş, kolay anlaşılır olmayan arayüzler ve standartlar ortaya çıkmıştır. Karşılaşılan bu sorunları çözmek üzere, var olan veya gelecekte var olacak modelleri birleştirecek bir yapı kurmak amacıyla 2001 yılında CMM Tümleştirme Projesi başlatılmıştır. 2002 yılında v1.0 olarak yayınlanan bu yeni modele Tümleşik Yetenek Olgunluk Modeli (Capability Maturity Model Integration – CMMI) adını verdiler. Daha sonra 2003 yılında CMMI v1.1, 2006 yılında CMMI v1.2 ve son olarak ta 2010 yılında CMMI v1.3 yayınlanmıştır.

CMMI v1.2 ile birlikte eski sürümde yer alan, CMMI-SW, CMMI-SE, CMMI-SS ve CMMI-IPPD modelleri kaldırılmış ve onların yerine “yıldız kümesi” adı verilen üç yeni oluşum getirilmiştir. Şekil 8'de görüldüğü üzere bu üç yeni oluşum; CMMI-DEV (Geliştirme amaçlı CMMI), CMMI-SVC (Hizmet amaçlı CMMI) ve CMMI-ACQ (Edinme amaçlı CMMI) olarak ifade edilmiştir. Özellikle kurumsal bilgi sistemi uygulamaları veya savunma sanayine yazılım geliştiren firmalar CMMI-DEV'i takip etmektedirler.



Şekil 8: CMMI yıldız kümesi

CMMI, yazılım geliştiren kuruluşlardaki süreç iyileştirme çabalarına rehberlik etmek ve yazılım işlerini gerçekleştirmek üzere mukavele yapılacak nitelikli şirketlerin belirlenmesini sağlamak amacıyla oluşturulmuştur (Yücalar, Şahinaslan, Borandağ, & Şahinaslan, 2010). CMMI tek model olmakla birlikte, Basamaklı Gösterim (Staged Representation) ve Sürekli Gösterim (Continuous Representation) olmak üzere iki gösterim şekli vardır (Chrissis, Konrad & Shrum, 2011).

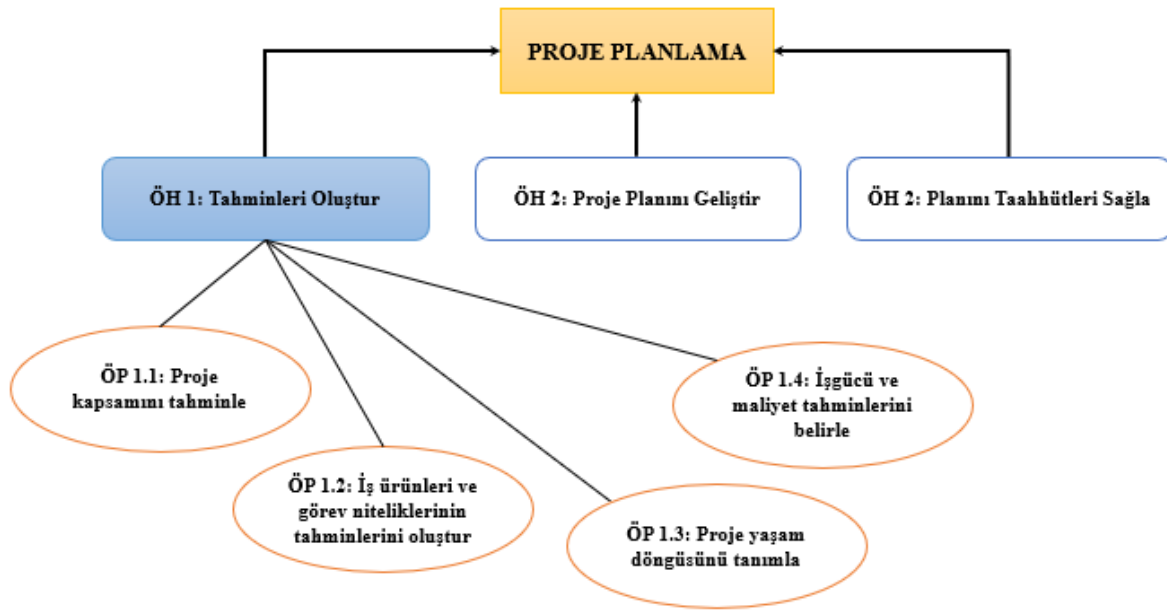
Basamaklı gösterim olgunluk düzeylerini tanımlarken, birçok süreç alanı karşısında bir organizasyonun süreç geliştirme başarısını kapsar. Bir kurum veya kuruluşun standart süreçlerinin olgunluğunu gösterir. 1 ile 5 arasında numaralandırılmış beş olgunluk seviyesi vardır. Sürekli gösterim yetenek düzeylerini tanımlarken, ayrı süreç alanları içinde bir organizasyonun süreç geliştirme başarısını kapsar. Tek bir süreç alanındaki (process area) iyileştirmeyi gösterir. 0 ile 5 arasında numaralandırılmış altı yetenek seviyesi vardır.

CMMI modelinin temelini süreç alanları oluşturmaktadır. Şekil 9’da görüldüğü üzere CMMI-DEV, teknoloji gelişimini artırmak için tasarlanmış 22 süreç alanından oluşmaktadır (Persse, 2007).



Şekil 9: CMMI-DEV süreç alanları (Persse, 2007)

Bu süreç alanlarından 16 tanesi temel modele dayanmaktadır. Geriye kalan 6 tanesi de teknoloji geliştirmenin özel ihtiyaçlarını karşılamaktadır. Bir süreç alanı, bir veya daha fazla özel hedefin bir araya getirilmesi ile oluşur. Bir süreç alanının tüm özel ve genel hedefleri yerine getirildiğinde o süreç alanı başarılı olur. Örneğin, Proje Planlama (Project Planning – PP) süreç alanının üç özel, bir genel hedefi vardır. Aynı zamanda bu özel ve genel hedeflerin yerine getirilmesi gereken özel ve genel pratikleri bulunmaktadır. Şekil 10’da proje planlama süreç alanının özel hedef ve pratikleri görülmektedir.



Şekil 10: Proje planlama süreci alanının özel hedef ve pratikleri

Süreç alanlarını düzenlemenin en iyi yolu, onları belirli kategorilere ayırmaktır. CMMI-DEV içerisinde yer alan 22 süreç alanı dört kategoriye ayrılmıştır. Aslında, bu zorunlu bir grupta değildir; farklı süreç alanlarını düşünmenin ve birbiriyle ilişkilendirmenin kolay bir yoludur. CMMI-DEV, sürekli gösterim olarak ele alındığında; Proje Yönetimi, Süreç Yönetimi, Mühendislik ve Destek olmak üzere dört süreç kategorisi vardır. Bu kategoriler içerisinde Proje Yönetimi süreç kategorisi içerisinde yer alan süreç alanları özellikle kurumsal bilgi sistemi uygulamalarının geliştirilmesinde dikkate alınmalıdır. Yazılım projelerinin yönetimi konusunda CMMI-DEV Proje Yönetimi süreç kategorisi proje yöneticilerine yol gösterecektir. Bu süreç alanlarının dikkate alınması proje başarısını da beraberinde getirecektir.

3.2.1.1 Proje Yönetimi Süreç Alanları

CMMI-DEV Proje Yönetimi Süreç Kategorisi altında, bir kuruluş içindeki çeşitli proje yönetimi ihtiyaçlarını doğrudan karşılamak için altı süreç alanı tanımlanmıştır. Bazıları diğerlerinden daha kademeli olarak bir sürecin yerine getirilmesini gerektirir. Ancak, projelerin nasıl planlandığına, izlendiğine, kontrol edildiğine ve ölçüldüğüne rehberlik etmek için hepsi aynı ana hedefle ilgilenmektedir. Proje Yönetim kategorisi altında altı süreç alanı yer almaktadır:

- Proje Planlama (Project Planning – PP),
- Proje İzleme ve Kontrol (Project Monitoring and Control – PMC),
- Risk Yönetimi (Risk Management – RSKM),
- Tedarikçi Anlaşma Yönetimi (Supplier Agreement Management – SAM),
- Tümüleşik Proje Yönetimi (Integrated Project Management – IPM),
- Nicel Proje Yönetimi (Quantitative Project Management – QPM).

Proje Yönetimi Süreç Kategorisi altında yer alan altı süreç alanının farklı renkte işaretlenmiş olduğu Şekil 9’da görülmektedir. Bu süreç alanlarını kısaca inceleyelim.

Proje Planlama:

Proje planlama süreci alanının amacı; proje etkinliklerini tanımlayan planları oluşturmak ve güncellemektir (Chrissis, Konrad & Shrum, 2011). İlave olarak, projeyi yönetmek için kullanılacak tahminlerin oluşturulmasını ele almaktadır. Proje planlama süreci alanı içerisinde; proje kapsamının

KURUMSAL BİLGİ SİSTEMİ UYGULAMALARI İÇİN POPÜLER PROJE YÖNETİM YAKLAŞIMLARI

belirlenmesi, iş ürünlerine ilişkin işgücü ve maliyet tahminlerinin yapılması, proje yaşam döngüsünün tanımlanması, bütçe ve proje takviminin belirlenmesi, proje risklerinin belirlenmesi, proje planının oluşturulması, oluşturulan planların gözden geçirilmesi ve plana ilişkin taahhüdün sağlanması gibi hususları ele almaktadır (Yücalar, 2006). Bu süreç alanı için tanımlanmış üç özel hedef vardır:

- ÖH 1: Tahminleri Oluştur (4 özel pratiği vardır)
- ÖH 2: Proje Planını Geliştir (7 özel pratiği vardır)
- ÖH 3: Plana İlişkin Taahhütleri Sağla (3 özel pratiği vardır)

Proje İzleme ve Kontrol:

Proje izleme ve kontrol süreç alanının amacı; proje planında herhangi bir sapma olduğunda yapılan düzeltici faaliyetlerin kontrol edilmesini ve proje gelişiminin takip edilmesini sağlamaktır (Chrissis, Konrad & Shrum, 2011). Bu süreç alanı Proje Planlama süreç alanının doğal bir uzantısıdır. Proje izleme ve kontrol süreç alanı; proje planlama parametrelerini izleme, taahhütleri ve riskleri izleme, veri yönetimini ve paydaş katılımını izleme, gelişme ve kilometre taşı gözden geçirmelerini uygulama, düzeltici faaliyetlerde bulunma ve yönetme gibi hususları içerir. Bu süreç alanı için tanımlanmış iki özel hedef vardır:

- ÖH 1: Plan Göre Projeyi İzle (7 özel pratiği vardır)
- ÖH 2: Düzeltici Faaliyetleri Kapanması için Yönet (3 özel pratiği vardır)

Risk Yönetimi:

Risk yönetimi süreç alanının amacı; ortaya çıkması muhtemel problemleri önceden belirlemektir (Chrissis, Konrad & Shrum, 2011). Aslında muhtemel riskler belirlenerek, bu risklerin proje veya ürün yaşam-döngüsündeki hedeflere ulaşılması noktasında ortaya çıkabilecek olumsuz durumların önüne geçilmiş olunacaktır. Bu süreç alanı; risk kaynaklarını belirleme, risk parametrelerini tanımlama, risk yönetim stratejisini oluşturma, riskleri belirleme ve analiz etme, riskleri değerlendirme, sınıflandırma ve risk önceliklerini belirleme, en önemli riskler için önleme (mitigation) planlarını geliştirme, riskleri izleme ve gerektiğinde önleme planlarını uygulama gibi hususları içerir. Bu süreç alanı için tanımlanmış üç özel hedef vardır:

- ÖH 1: Risk Yönetimine Hazırlık (3 özel pratiği vardır)
- ÖH 2: Riskleri Belirleme ve Analiz Etme (2 özel pratiği vardır)
- ÖH 3: Riskleri Önle (2 özel pratiği vardır)

Tedarikçi Anlaşma Yönetimi:

Tedarikçi anlaşma yönetimi süreç alanının amacı; yapılan resmi anlaşmalar doğrultusunda tedarikçilerden ürün satın almayı yönetmektir (Chrissis, Konrad & Shrum, 2011). Ayrıca, proje ekibinin tedarikçi anlaşmalarını nasıl oluşturabileceğini ve daha sonra verimli bir entegrasyon sağlamak için tedarikçilerle olan çalışmaların nasıl koordine edilebileceğini de ele almaktadır (Persse, 2007). Bu süreç alanı; ürün ve ürün bileşenlerinin satın alınmasında kullanılacak yöntemin belirlenmesi, tedarikçi seçimi, aday COTS (Commercial Of The Shelf) ürünlerini değerlendirme, anlaşmaları oluşturma, güncelleme ve işletme, ürünleri kabul etme ve yeni ürüne geçiş sağlama ile ilgili hususları içerir. Bu süreç alanı için tanımlanmış iki özel hedef vardır:

- ÖH 1: Tedarikçi Anlaşmalarını Oluştur (3 özel pratiği vardır)
- ÖH 2: Tedarikçi Anlaşmasını Sağla (4 özel pratiği vardır)

Tümleşik Proje Yönetimi:

Tümleşik proje yönetimi süreç alanının amacı; projeyi ve ilgili paydaşların katılımını standart süreçlerden uyarlanan bütünleşik ve tanımlı bir süreç altyapısı ile yönetmektir (Chrissis, Konrad & Shrum, 2011). Ayrıca, tümleşik ürün ve süreç geliştirme proje ve proje hedefleri için ortak bir vizyon ve ekip yapısı oluşturmaktır. Bu süreç alanı; projeye ilişkin tanımlı süreçleri oluşturma,

KURUMSAL BİLGİ SİSTEMİ UYGULAMALARI İÇİN POPÜLER PROJE YÖNETİM YAKLAŞIMLARI

organizasyonel süreç varlıklarını kullanma, planları birleştirme, projeyi birleştirilmiş planlar ile yönetme, paydaşların katılımını ve bağımlılıklarını yönetme, koordinasyonu sağlama gibi hususları içerir. Bu süreç alanı için tanımlanmış üç özel hedef vardır:

- ÖH 1: Tanımlı Süreçleri Oluştur (6 özel pratiği vardır)
- ÖH 2: İlgili Paydaşlarla İşbirliği ve Koordinasyon (3 özel pratiği vardır)
- ÖH 3: Tümüleşik Ürün ve Süreç Geliştirme İlkelerini Uygula (5 özel pratiği vardır)

Nicel Proje Yönetimi:

Nicel proje yönetimi süreç alanının amacı; projenin belirlenen kalite ve süreç-performans hedeflerine ulaşması için projenin tanımlanmış sürecini nicel (sayısal) olarak yönetmektir (Chrissis, Konrad & Shrum, 2011). Bu süreç alanı; proje hedeflerini oluşturma, geçmiş verileri dikkate alarak süreci oluşturma, istatistiksel olarak yönetilecek alt süreçleri seçme, proje performansını hedeflere göre yönetme, alt süreç performansını yönetme, istatistiksel verileri kurumun ölçüm veri tabanına kaydetme gibi hususları içerir. Bu süreç alanı için tanımlanmış iki özel hedef vardır:

- ÖH 1: Projeyi Nicel Olarak Yönet (4 özel pratiği vardır)
- ÖH 2: Alt Süreç Performansını İstatistiksel Olarak Yönet (4 özel pratiği vardır)

4. PROJE YÖNETİMİNDE BAŞARILI OLMAK İÇİN ÖNERİLER

Proje geliştirme süreci zaman alan ve birçok problemle karşılaşılan bir süreçtir. Projelerin başarılı olması için proje yönetim sürecine gereken önemin verilmesi gerekmektedir. Proje yönetiminde başarıyı yakalayabilmek için doğru yöntem veya yaklaşımın seçilmesi son derece önemlidir. Proje yönetim sürecinde Scrum, XP, Kanban gibi bir çevik yaklaşım ya da CMMI gibi bir disiplinli model tercih edilebilir. Aslında proje yönetim sürecinde başarıyı garanti etmek adına hem bir çevik yöntemden hem de bir disiplinli modelin kullanımından yararlanılabilir. Çünkü hem çevik yöntemler hem de CMMI gibi disiplinli modeller yapı itibarı ile Deming Döngüsüne (Planla-Yap-Kontrol Et-Harekete Geç) dayanırlar ve bu nedenle birkaç değeri birlikte paylaşırlar. En basit ifade ile, çevik metodolojiler NASIL yapılması gerektiğini belirtirken CMMI gibi disiplinli modeller NE yapılması gerektiğini belirtirler. Disiplinli modeller ile çevik yöntemler arasındaki görünen fikir ayrılığı ise, çevik yöntemlerin doğrudan ürüne katkıda bulunan çıktılara odaklanıyor olmasından kaynaklanmaktadır. Disiplinli yöntemler, ürüne doğrudan katkısı olmayan teslimatları israf olarak nitelendirir. Görünen fikir ayrılığının ele alınacağı en iyi perspektif risk yönetimidir. Her iki yaklaşımda da ele alınan projeler, riski en aza indirecek şekilde yönetilmeye çalışılır. Ancak, çevik yöntemlerde ürün bileşenleri için farklı çevik ölçeklendirme yapılarının kullanımı örgütsel düzeyde riski yönetmeyi zorlaştırır. CMMI'da ise bu durum nicel proje yönetimi ve risk yönetimi süreç alanlarının kullanımı ile aşılar. Her iki yöntemin birlikte kullanımında optimum dengeyi sağlamak proje başarısını beraberinde getirecektir. Proje yönetiminde çevik yöntemlerin önermiş olduğu temel pratiklere karşılık CMMI'nin süreç alanlarının dikkate alınması faydalı olacaktır.

5. KAYNAKÇA

Aston, B. (2019). 9 of the Most Popular Project Management Methodologies Made Simple. (Retrieved February 5, 2020), from <https://thedigitalprojectmanager.com/project-management-methodologies-made-simple/> website.

Beck, K. (2004). Extreme Programming Explained: Embrace Change (2nd Edition). Addison-Wesley.

Borandağ, E., Yücalar, F., Erdoğan, Ş. Z. & İnce, F. (2009). Basamaklı CMMI Modeli ile Extreme Programming Metodunun Değerlendirilmesi. Trakya Üniversitesi Fen Bilimleri Dergisi. 10(1), 61-67. ISSN: 1305 6468.

KURUMSAL BİLGİ SİSTEMİ UYGULAMALARI İÇİN POPÜLER PROJE YÖNETİM YAKLAŞIMLARI

Brooks, F. P. (1987). No Silver Bullet Essence and Accidents of Software Engineering. Computer, IEEE. 20(4), 10-19. doi:10.1109/MC.1987.1663532

Çamoğlu, K., Akbayır, D., Yücalar, F., Bayraklı, S. (2010). Bir Çevik Yazılım Geliştirme Sürecinin Uyarlanması ve Uygulanması. Havacılık ve Uzay Teknolojileri Dergisi. 4(3), 57-67.

Chrissis, M. B., Konrad, M. & Shrum, S. (2011). CMMI for Development: Guidelines for Process Integration and Product Improvement. 3rd Edition. Addison-Wesley Professional.

Cohn, M. (2010). Succeeding with Agile: Software Development Using Scrum (1st Edition). Addison-Wesley Professional.

Değerli, M., Özbudak, E. (2017). Yazılım Mühendisliği ve Sistem Mühendisliği Süreçlerinin Harmanlanması: ISO/IEC 12207:2008'in Getirdikleri ve Değişikliklerin Etkilerine Yönelik Değerlendirmeler. 11. Ulusal Yazılım Mühendisliği Sempozyumu (UYMS'17), 368-377. Alanya, Türkiye.

Erdoğmuş, U. (2009). Süreç İyileştirmede CMMI Modelleri ve Türkiye'de CMMI Uygulamalarının Durumu. Yüksek Lisans Tezi. İstanbul Teknik Üniversitesi. Fen Bilimleri Enstitüsü. Yüksek Lisans Tezi.

Extreme Programming: A gentle introduction. (Retrieved February 3, 2020), from extremeprogramming.org website, <http://www.extremeprogramming.org/>

Highsmith, J. (2002). Agile Software Development Ecosystems. 1st Edition. Addison-Wesley Professional.

Hiranabe, K. (2008). Kanban applied to software development: From agile to lean. (Retrieved February 5, 2020). from infoq.com website, <https://www.infoq.com/articles/hiranabe-lean-agile-kanban/>

Hughes, B., Cotterel, M. (2009). Software Project Management (5th Edition). McGraw-Hill.

Kruchten, P. (2003). The Rational Unified Process: An Introduction (3rd Edition). Addison-Wesley Professional.

Manifesto for Agile Software Development, (Retrieved February 2, 2020), from agilemanifesto.org website, <http://agilemanifesto.org>

Miletić, M., Miletić, I. (2017). Lean Methodology and Its Derivates Usage for Production Systems in Modern Industry. Applied Engineering Letters. 2(4), 144-148.

Persse, J. (2007). Project Management Success with CMMI: Seven CMMI Process Areas (1st Edition). Prentice Hall.

Pichler, R. (2010). Agile Product Management with Scrum: Creating Products that Customers Love (1st Edition). Addison-Wesley Signature Series.

Principles Behind the Agile Manifesto. (Retrieved February 2, 2020), from agilemanifesto.org website, <http://agilemanifesto.org/principles.html>

Project Management Basics. (Retrieved February 1, 2020). from <https://www.wrike.com> website, <https://www.wrike.com/project-management-guide/project-management-basics/>

Raut, L.P., Wakode, R.B., Talmale, P. (2015). Overview on Kanban Methodology and its Implementation. International Journal for Scientific Research & Development. 3(2), 2518-2521.

Rout, T.P. (2004). ISO/IEC 15504 – Evolution to an International Standard. Software Process: Improvement and Practice. 8(1), 27–40. doi:10.1002/spip.167

Schach, S. R. (2010). Object-Oriented and Classical Software Engineering. 8th Edition. McGraw Hill.

Schwaber, K. (2004). Agile Project Management with Scrum (1st Edition). Microsoft Press.

KURUMSAL BİLGİ SİSTEMİ UYGULAMALARI İÇİN POPÜLER PROJE YÖNETİM YAKLAŞIMLARI

Siderova, S. (2019). The Kanban Way: How To Visualize Progress And Data In Trello. (Retrieved February 5, 2020), from website <https://blog.trello.com/kanban-data-nave>

Stellman, A., Greene, J. (2013). Learning Agile: Understanding Scrum, XP, Lean, and Kanban. 1st Edition. O'Reilly Media.

Tapsız, E. B. (2018). Proje Yönetimi ve Planlama Süreçleri. İstanbul İşletme Enstitüsü web sayfasından alındı (Retrieved February 1, 2020). <https://www.iienstitu.com/blog/proje-yonetimi-ve-planlama-surecleri>

Yücalar, F. (2006). Süreç Odaklı Kalite Yönetimi Anlayışına Hâkim Yazılım Sektöründeki Firmaların CMMI Basamaklı Modeli ile Değerlendirilmesi. Yüksek Lisans Tezi. Maltepe Üniversitesi Fen Bilimleri Enstitüsü. İstanbul.

Yücalar, F., Borandağ, E. (2018). Yazılım Mühendisliğinde Modern Yaklaşımlar (1. Basım). Nobel Akademik Yayıncılık. Ankara, Türkiye.

Yücalar, F., Şahinaslan, E., Borandağ, E. & Şahinaslan, Ö. (2010). Yazılım Yöneticileri İçin Tümüleşik Yetenek Olgunluk Modeli: Genel Bir Bakış. XII. Akademik Bilişim Konferansı (Ab2010). (10-12 Şubat 2010). Muğla Üniversitesi. Muğla, Türkiye.

MEME KANSERLERİNİN TEŞHİSİNDE KULLANILAN GÖRÜNTÜ İŞLEME TEKNİKLERİYLE İLGİLİ BİR DERLEME ÇALIŞMASI

Güliz Toz *, Pakize Erdoğan

** Karamanoğlu Mehmetbey Üniversitesi, Ermenek, Karaman*

glz.toz@gmail.com

** Sorumlu Yazar*

1. GİRİŞ

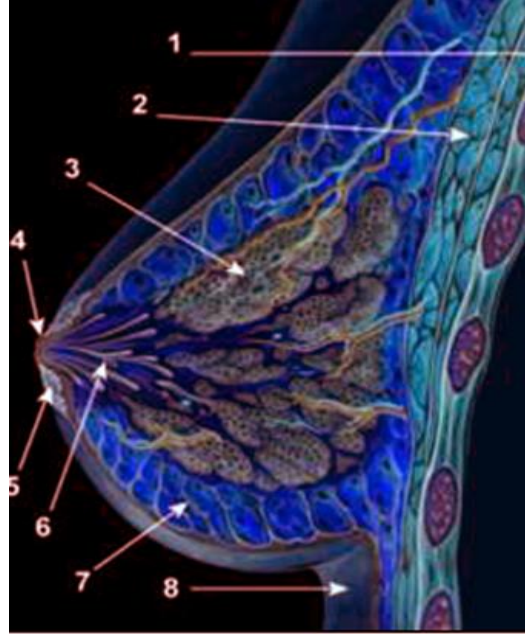
Meme kanseri dünyada ve ülkemizde, kadınlarda görülen kanser türleri açısından ön sıralarda yer almakta ve çoğunlukla ölümlü sonuçlanmaktadır. Günümüzde meme kanserinin teşhisine yönelik pek çok çalışma bulunmaktadır. Hastalığın erken teşhisi, ölüm oranını etkili bir şekilde azaltabilmektedir. Bu nedenle hastalığın erken teşhisine yönelik yapılan çalışmalar büyük önem arz etmektedir. Bu kanserin temel belirtileri meme bölgesinde bulunan mikrokalsifikasyonlar, ikili asimetri, kitleler ve mimari bozulmadır. Meme Bölgesinde yer alan bu anomaliler farklı yöntemlerle teşhis edilebilmektedir. Bu yöntemler; mamogram, ultrason, MRI ve biyopsi olarak sıralanabilir. Bunlardan en sık kullanılanı mamogram yöntemi olmakla birlikte, vakanın durumuna göre ultrason, MRI ve gerekli görüldüğü durumlarda biyopsi yöntemleri de tercih edilebilmektedir. Bunların dışında Bilgisayar Destekli Tanı (BDT) da makine öğrenmesinde ve yapay zekâ tekniklerindeki gelişmeler sayesinde meme kanserinin teşhisine yardımcı olacak bir sistem olarak kullanılabilir. Meme kanserine yönelik BDT sistemi temel olarak bu kanserin teşhisinde kullanılan görüntüleme sonuçlarının radyoloğun daha kolay yorumlamasına olanak sağlayacak şekilde düzenlenmesini sağlamaktır. BDT sistemi, radyoloji uzmanına ikinci bir göz olarak yardımcı olan uygun maliyetli alternatif bir yöntemdir. BDT sisteminin aşamalarına bakıldığında genel olarak dört ana başlık altında toplandığı görülmektedir. Bunlar; önışleme, bölütleme, öznelik çıkarımı ve sınıflandırma aşamasıdır. Önışleme aşamasında çoğunlukla gürültünün kaldırılması, yumuşatma, görüntü iyileştirme, filtreleme ve histogram eşitleme gibi amaca en uygun başlangıç görüntüsüne ulaşmayı sağlayan işlemler gerçekleştirilmektedir. Bölütleme aşaması, görüntüde olan-olması muhtemel anomalilerin, görüntünün ilgisiz bölgelerinden ayrılması amacıyla yapılan çalışmaları kapsamaktadır. Özneliklerin elde edilmesi ise bölütleme aşamasında elde edilen şüpheli bölgelerin farklı özelliklerinin elde edilmesine yönelik çalışmalardan oluşmaktadır. Son olarak sınıflandırma aşaması elde edilen özneliklerin genel olarak kanserli, kansersiz ve-veya iyi huylu kötü huylu olarak tanı konulduğu aşamadır. Literatürde bu amaçla geliştirilmiş birçok çalışma yer almaktadır. Bu bölümde sunulan çalışmanın amacı da literatürde yer alan bu çalışmaları ele almak ve meme kanserleri konusunda çalışmak isteyen araştırmacılara bir derleme sunmaktır. Bu amaçlarla bu bölümde ilk olarak meme anatomisi tanıtılmış daha sonra meme kanseri ile ilgili bazı istatistiksel bilgilere yer verilmiş ardından meme görüntüleme yöntemlerinden en çok kullanılanlardan biri olan mamografi tanıtılmış ve son olarak bilgisayar destekli teşhis sistemleri hakkında literatürde yer alan çalışmalar, BDT sistem aşamalarına göre derlenmiştir, böylece konu ile ilgili araştırmacılara kendi BDT sistemlerinin her bir aşamasında hangi yöntemi kullanmaları konusunda fikir vermek amaçlanmıştır. Bu çalışma Güliz Toz'un (Toz G. (2018)), doktora tezine dayanmaktadır.

2. MEME ANATOMİSİ

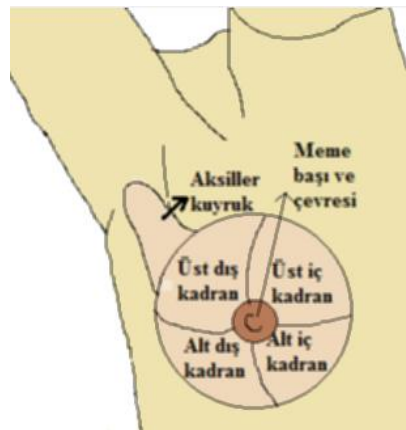
Erişkin kadın memesi, göğüs ön duvarındaki göğüs kaslarının üstünde bulunan ve orta hatta göğüs kemiğinin dış kenarına, yan hatta koltuk altının ön sınırından aşağı doğru uzanan, süt üreten bezlerden oluşan göğüs duvarına bağlı damla şeklinde bir çift organdır(Özmen ve diğerleri (2012)). Meme yapısı, göğüs duvarının en büyük kasları olan pektoral kasların üzerine inşa edilmiştir ve kendi yapısında kas dokusu yoktur(Özmen ve diğerleri (2012)). Her bir memede, etrafı yağ dokusuyla çevrili olan ve içlerinde birçok lobül bulunan 15- 20 lop bulunmaktadır (Özmen ve diğerleri (2012)). Lopların içinde süt üretimi yapan süt bezlerine bulunmaktadır(Özmen ve diğerleri (2012)). Büyük süt kanalları, süt bezlerinden çıkan süt kanallarının birleşmesiyle meydana gelir(Özmen ve diğerleri (2012)). Meme başının çevresinde yer alan pigmentli alana areola denir(Özmen ve diğerleri (2012)). Memenin anatomik yapısı Şekil 2.1’de sunulmuştur.

Meme bölgesi, hastalıklarının yerini belirtmek amacıyla kadrana ayrılmıştır. Her bir kadranın yer aldığı görüntü Şekil 2.2’de verilmiştir. Koltuk altı kuyruğu (aksiller kuyruk), meme dokusunun bazı durumlarda koltuk altına kadar uzanan kısmına verilen isimdir. Şekil 2.2’de bu kuyruk görülebilmektedir. Memenin şekli ve büyüklüğü kişiden kişiye hatta aynı kişide her iki memede bile değişiklik gösterebilmektedir (Ateş S. (2014)).

1. Göğüs duvarı
2. Pektoral kas
3. Lobüller
4. Meme başı
5. Areola
6. Laktiferöz kanal
7. Yağ dokusu
8. Cilt



Şekil 2.1 Memenin anatomik yapısı (Ateş S. (2014)).



Şekil 2.2 Memenin dış görünümü ve kadrantları (Ateş S. (2014)).

3. MEME KANSERİ VE İSTATİSTİKLER

Bir doku veya organdaki hücrelerin kontrol dışı bölünüp çoğalmasıyla oluşan kötü urlara kanser denir. Normalde ölen hücrelerin yenilenmesi ve yaralanan dokuların onarılması için hücrelerin büyümesi ve bölünmesi gerekir. Fakat bazen bu süreç yanlış işler, yeni hücrelere gerek olmadan hücreler kontrolsüz bölünmeye başlar ve çoğalırlar. Kontrolsüz çoğalan bu hücreler tümörü oluştururlar ve bu tümörler iyi huylu veya kötü huylu olarak sınıflandırılırlar. İyi huylu tümörler genellikle alınırlar ve alındıktan sonra tekrarlamazlar. Kanser olmayan tümör olarak sınıflandırılırlar. Kötü huylu tümörler ise kanser olarak sınıflandırılırlar böyle tümörlerde hücreler kontrolsüz ve düzensiz bölünürler. Bu tümörler anormaldirler, normal dokuların içine sızabilir ya da dokuları tahrip edebilirler. Tedavi edilmemesi durumunda ciddi rahatsızlıklara yol açabilirler, hatta ölüme neden olabilirler (“Halk Sağlığı Genel Müdürlüğü”).

Meme kanseri de meme dokusundaki hücrelerde meydana gelen kanserlerdir. En çok görülen türü “duktal” kanser olarak anılan meme kanallarından kaynaklanan kanser türüdür (“Halk Sağlığı Genel Müdürlüğü”). İkinci sırada ise “lobüler” kanserler yer almaktadır ve süt üreten bezlerden kaynaklanır. Ayrıca daha az rastlanan medüller, tübüler, müsinöz gibi türleri de vardır (“Halk Sağlığı Genel Müdürlüğü”). Mamografik değerlendirme için meme lezyonlarının doğru ve standart bir şekilde tanımlanması, malign patoloji olasılığının belirlenmesi, uzmana mamografi sonucu olarak net bir mesaj verilmesi amacıyla Amerikan Radyoloji Koleji (ACR) tarafından 1993 yılında ‘Meme Görüntüleme Raporlama ve Veri Sistemi’ (Breast Imaging Reporting and Data System -BIRADS) adı altında bir raporlama sistemi geliştirilmiştir (American College of Radiology (ACR)(1998)). Bu sisteme göre memede karşılaşılan sorunların sınıflandırılması ve raporlama sistemi standart hale getirilmiştir. BIRADS sınıflandırmayı 6 kategoride yapmaktadır. Bunlar;

Kategori 0: Yetersiz tetkik.

Kategori I: Negatif. Memeler simetrik, kitle yok, şüpheli kalsifikasyon yok.

Kategori II: İyi huylu bulgular.

Kategori III: Muhtemelen iyi huylu, kısa aralıkla kontrol önerilir.

Kategori IV: Kötü huylu olma şüphesi. Biyopsi önerilir.

Kategori IV a: Kötü huylu olma şüphesi düşük

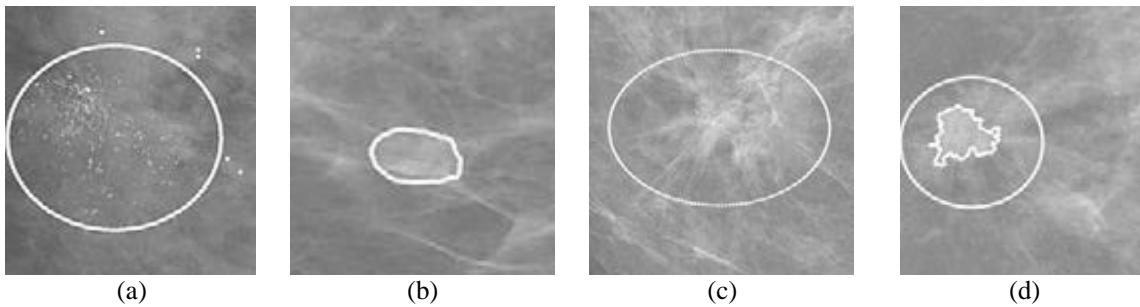
Kategori IV b: Kötü huylu olma için ara şüphe

Kategori IV c: Kötü huylu olma şüphesi orta

Kategori V: Kötü huylu olma şüphesi kuvvetli. Gerekli işlemler yapılmalıdır.

Kategori VI: Bilinen (biyopside kanıtlanmış) kötü huylu tümör (American College of Radiology (ACR)(1998)).

Memede karşılaşılan kitle türlerine, kalsifikasyon, iyi tanımlanmış kitle, spiküle kitle ve mimari bozulma örnek verilebilir. Bu örneklere ait görüntüler Şekil 3.1’de verilmiştir.



Şekil 3.1 Kitle örnekleri (a) Kalsifikasyon (b) İyi tanımlanmış kitle (c) Mimari bozulma (d) Spiküle kitle (Moreira ve diğerleri (2012))

MEME KANSERLERİNİN TEŞHİSİNDE KULLANILAN GÖRÜNTÜ İŞLEME TEKNİKLERİYLE İLGİLİ BİR DERLEME ÇALIŞMASI

Yakın akrabada meme kanseri görülmüş olması, yaşlanma ve yoğun meme dokusuna sahip olunması meme kanseri için yüksek derecede risk oluştururken boy ve kilo, ilk adet yaşı, menopoz sonrası hormon tedavisi, doğum kontrol hapları, hiç doğum yapmamış olmak, ilk doğum yaşı, emzirme, menopoz yaşı, hareketsizlik, proliferatif meme hastalıkları, beslenme ve alkol kullanımı da riskli durumlar olarak karşımıza çıkmaktadır (Güllüoğlu B. M.(2008)).

Meme kanserinin erken evrede tespit edilmesi ölüm oranlarının azaltılması ve tedavi süreci açısından oldukça önemlidir. Erken tanı amacıyla doktor tarafından yılda bir yapılan meme muayeneleri, kişinin kendi kendine yaptığı meme kontrolleri ve meme görüntüleme yöntemleri kullanılmaktadır(“Halk Sağlığı Genel Müdürlüğü”).

Meme kanseri, kadınlarda görülen tüm kanserler arasında, Amerikan Kanser Derneği'nin (American Cancer Society) 2012 verilerine göre yaklaşık %30'luk bölümünü oluşturmaktadır (Ateş S. (2014)). Ülkemizde ise bu oran %25,6'dır. Türkiye'de kanserden kaynaklanan ölümlere bakıldığında meme kanseri en sık ölüm nedeni olarak belirtilmektedir (%17,6) (Özmen V. (2013)). Türkiye'de meme kanseri sıklığına bakıldığında IARC Globocan 2008 verilerine göre %28,3'lük bir oran görülmektedir(Özmen V. (2013)). Dünya Sağlık Örgütü'nün (WHO) bildirdiğine göre her yıl dünyada 1,2 milyon kişiye meme kanseri tanısı konulmakta ve bunların 500 000 kadarı bu hastalık yüzünden hayatını kaybetmektedir(Ateş S. (2014)) 2008 yılı için ABD genelinde yeni meme kanseri vaka sayısı kadınlarda 182.460, erkeklerde 1990 vaka olarak hesaplanırken; beklenen ölümler ise kadınlarda 40.480, erkeklerde 450 vaka olarak National Cancer Institute (NCI), tarafından rapor edilmiştir. NCI 2000-2004 verilerine bakıldığında hayatı boyunca her sekiz kadından birinde meme kanseri gelişebileceği ve her otuz kadından birinin meme kanseri nedeniyle hayatını kaybedeceği tahmin edilmektedir (Ateş S. (2014)).

4. MAMOGRAFİ

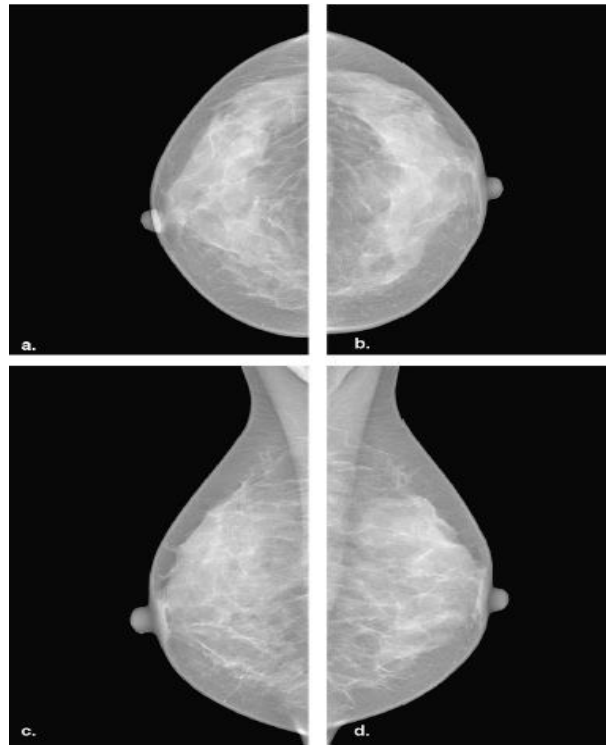
Radyologlar meme hastalıklarının tanısında ve meme kanserinin erken saptanmasında mamografi, ultrasonografi, bilgisayarlı tomografi ve manyetik rezonans görüntüleme (MRI) gibi farklı görüntüleme yöntemleri kullanılmaktadır. Bunlardan mamografi günümüzde yaygın olarak kullanılan etkinliği kanıtlanmış temel meme görüntüleme yöntemidir. Mamografinin sıklıkla tercih edilmesinin nedeni, düşük radyasyon dozunda kaliteli görüntüleme performansı sağlayabilmesi ve bunu düşük maliyetlerle yapabilmesidir. Mamografi işlemi esnasında, meme sıkıştırılarak kalınlığı homojen bir şekilde azaltılmış olur böylece saçılan radyasyonun azaltılması, görüntü keskinliğinin artırılması sağlanır(Gedik N.(2013)). Ayrıca, meme sabit olduğundan harekete bağlı görüntü bulanıklığı en aza indirgenir(Gedik N.(2013)).Görüntülenen meme dokusu miktarı artar, nesne ve reseptör mesafesi kısaltılarak geometrik bulanıklığın azaltılmasını sağlar(Gedik N.(2013)).Sıkıştırma sistemi ince sert plastikten yapılan kompresyon plağı ve kontrol mekanizmasından oluşur(Gedik N.(2013)).Meme kanserininin mamografi ile saptanma duyarlılığı % 63–98 oranında iken bu oran meme dokusu yoğun olan memelerde %30-48'lere düşmektedir(Ateş S. (2014)). Dolayısıyla meme dokusu yoğun vakalarda ultrasonografi ve manyetik rezonans görüntüleme (MRG) gibi ek tanı yöntemlerinde gerekli olduğu belirtilmektedir (Ateş S. (2014)). Dijital mamografi cihazı ve görüntüleme birimi şekil 4.1'de verilmiştir.

MEME KANSERLERİNİN TEŞHİSİNDE KULLANILAN GÖRÜNTÜ İŞLEME TEKNİKLERİYLE İLGİLİ BİR DERLEME ÇALIŞMASI



Şekil 4.1 Dijital Mamografi Cihazı ve Görüntüleme Birimi(Özmen V. (2013)).

Mamografi incelemelerini tarama ve tanı amaçlı olmak üzere iki gruba ayırılabilir. Tarama amaçlı mamografi hastalığın erken teşhisi amacıyla 40 yaş ve üzeri risk grubunda olan kadınlara hastalık ortaya çıkmadan önce yapılır. Erken teşhis tedavinin etkili olması ve kanser nedeniyle gerçekleşen ölümlerin azaltılmasında oldukça önemlidir(Özmen V. (2013)). Tarama amaçlı mamografi elde edilirken meme içinde anormal yapıların fark edilebilmesi amacıyla iki farklı yönde görüntü elde edilir. Bunlar her iki meme için mediolateral eğik görünümü (MLO) ve kraniokaudal (CC) görünümüdür (Şekil 4.2) (Moreira ve diğerleri (2012))



Şekil 4.2 Dijital mamogram örnekleri:(a) sağ memenin kraniokaudal (CC) görünümü; (b) sol memenin CC görünümü ; (c) sağ memenin mediolateral eğik görünümü (MLO); (d) sol memenin MLO görünümü(Moreira ve diğerleri (2012))

5. BİLGİSAYAR DESTEKLİ TEŞHİS SİSTEMLERİ

Mamografi sıklıkla tercih edilen ve meme kanserinin teşhisinde etkin bir görüntüleme yöntemidir. Meme kanserlerinin yaklaşık %85-90'ı mamografi incelemeleri sonucu elde edilirken, görüntülenen mamografilerdeki tümörlerin yaklaşık %10-15'i çeşitli sebeplerden radyologlar tarafından gözden kaçırılmaktadır (Dheeba, Albert, Tamil(2014)). Bu sebepler radyolojik anormalliklerin yaygınlığı, dikkati kaybetme ve yüksek miktarda mamografinin okunduğu taramalarda radyoloğun yorgunluğu olarak tanımlanabilir (Ciatto ve diğerleri (2006)). Özellikle meme dokusu yoğun ve/veya genç kadınlarda kitlelerin teşhisi daha da güçleşmektedir (Kurt B.(2014)). Bilgisayar Destekli Teşhis Sistemlerinin temel amacı anormal yapıları ön plana çıkarmak ve radyoloğun dikkatini şüpheli bölgelere çekmektir. Böylece radyologlara " ikinci bir okuyucu " olarak yardımcı olabilmektir (Ciatto ve diğerleri (2006)). BDT sistemleri ile elde edilen şüpheli bölgelerin görsel olarak kontrolü radyolog tarafından yapılır, radyolog eğitim ve tecrübelerine dayanarak nihai kararı verir (Ciatto ve diğerleri (2006)).

Bu bölümün devamında bu konuda yapılmış literatür çalışmalarına yer verilmiştir.

Meme kanserinin bilgisayar destekli yöntemlerle tespiti ve teşhisine yönelik çalışmalar; önişleme, bölütleme, öznetelik çıkarımı ve sınıflandırma olmak üzere temel olarak dört gruba ayrılabilir.

5.1. Önişleme Aşaması

Önişleme aşamasında çoğunlukla gürültünün kaldırılması, yumuşatma, görüntü iyileştirme, filtreleme ve histogram eşitleme gibi amaca en uygun başlangıç görüntüsüne ulaşmayı sağlayan işlemler gerçekleştirilmektedir (Gowri ve Amudha (2014)).

5.1.1. Gürültü Giderme

Görüntü üzerinde farklı nedenlerden dolayı oluşan istenmeyen sinyaller gürültü olarak tanımlanabilmektedir. Gürültüyü oluşturan farklı nedenler bulunduğundan çözüm için de yine bu nedenlere çözüm olabilecek farklı algoritmalar geliştirilmiştir. Gürültü giderme algoritmalarından filtreleme sıklıkla kullanılan bir algoritma olup temel olarak alçak geçiren ve yüksek geçiren olmak üzere ikiye ayrılır. Bununla birlikte filtreler doğrusal olan veya olmayan, zamanla değişen veya değişmeyen, analog veya dijital, ayrık veya sürekli, pasif veya aktif şekilde olabilmektedir (Gowri ve Amudha (2014)). Yumuşatma işlemi de yine gürültünün azaltılması, daha az bulanıklık ve görüntü içindeki gereksiz piksellerin kaldırılmasıyla ilgili işlemlerdir (Gowri ve Amudha, (2014)). Bu amaçla yapılan çalışmalara bakıldığında bir çok farklı filtre kullanıldığı görülmektedir. Vijikala ve Dhas (Vijikala ve Dhas (2016)) mamogram görüntülerinde gürültü giderilmesi amacıyla Hibrid Medyan Filtresi (HMF), Yönlü Rician Gürültü Azaltma Anizotropik Difüzyon (ORNRAD), Yüksek Mertebeden Hibrit Ortalama (HOHM) ve Yerel Olmayan Ortalama Seviye (NLM) filtrelerini kullanmışlar ve filtrelerin performans analizi için Ortalama Kare Hata (MSE), Doğrusal Minimum Ortalama-Kare Hata (LMMSE), En Yüksek Sinyal Gürültü Oranı (PSNR), Görüntü Kalitesi İndeksi (IQI), Ortalama Mutlak Hata (MAE) ve Kontrast-Gürültü Oranı (CNR) parametrelerini kullanmışlardır. Bir diğer çalışma ise (Kumar ve diğerleri (2016)), mekansal ve frekans alanında gürültü giderme amacıyla kullanılan uyarlamalı histogram eşitlemesi, medyan filtre, butterworth filtre, frost filtre ve dalgacık filtre olmak üzere beş algoritmadan hangisinin mamografi görüntülerine daha uygun olduğunu bulmak için yapılmıştır. Bu gürültü giderme tekniklerini 5 iyi huylu ve 5 kötü huylu vakadan oluşan 10 görüntü üzerinde test etmişler ve sonuçlar uzaysal alan için medyan filtrenin ve frekans alanı için dalgacık ile gürültü gidermenin daha yüksek PSNR (sinyal gürültü oranı) ve düşük MSE (ortalama karekök hata), MD (maksimum fark), NAE (normalleştirilmiş mutlak hata) ve SC Yapısal içerik) değerlerine sahip olduğunu göstermiştir. Bir diğer çalışmada ise Mencattini ve arkadaşları (Mencattini ve diğerleri(2008)) mamogramlarda gürültü giderilmesi amacıyla diyadik wavelet dönüşümüne dayanan yeni bir algoritma sunmuşlardır. Gürültü giderme aşaması yerel yinelemeli gürültü varyans tahminine dayanmaktadır. Ayrıca, mikro kalsifikasyonlar için, farklı dalgacık ölçeklerinde adaptasyonlu bir uyum derecesi ayarlaması önermişlerdir; buna karşın kitle

MEME KANSERLERİNİN TEŞHİSİNDE KULLANILAN GÖRÜNTÜ İŞLEME TEKNİKLERİYLE İLGİLİ BİR DERLEME ÇALIŞMASI

tespiti durumunda, diyadik dalgacık bilgilerini matematiksel morfoloji ile birleştiren yeni bir segmentasyon yöntemi geliştirmişlerdir. Önerilen algoritma, hem analitik indeksler hem de radyologların görüşleri ile literatürde önerilen birkaç algoritma ile elde edilen sonuçlarla karşılaştırarak, çok sayıda klinik görüntü üzerinde test edilmiştir. Ön testlerden elde edilen sonuçlar, yöntemin diğer yaklaşımlara oranla meme kanserinin erken teşhisinde anlamlı derecede etkili olduğunu göstermiştir. Hamed (Hamed ve diğerleri (2018)) ve arkadaşları çalışmalarında Mini-MIAS veritabanından aldıkları dijital mamogramlarda gürültü giderme amacıyla en çok kullanılan wiener, ortalama, medyan, adaptif medyan ve Gauss filtrelerini uygulamışlardır. Elde ettikleri verileri ortalama kare hata, PSNR ve bulanıklık ölçütlerine göre değerlendirmişlerdir. Deneysel sonuçlara bakıldığında dijital mamogramlarda tuz ve biber gürültüsünü gidermek amacıyla en iyi sonucu adaptif medyan filtre vermiştir. Ortalama filtre ise yüksek bulanıklık değeriyle birlikte Gauss, benek ve poisson gürültüsünü gidermede en iyi sonucu vermiştir. Bu sorunun üstesinden gelmek amacıyla da wiener filtre kullanmışlardır. Yapılan çalışmada farklı gürültü türleri için en iyi filtreler birlikte kullanılmış ve en düşük ortalama kare hatanın elde edildiği kombinasyonda 322 mamogram için ortalama kare hatada 11.3106 değeri, ortalama PSNR için ise 37.9023 değeri elde edilmiştir.

5.1.2. Görüntü İyileştirme

Önişleme yöntemlerinden görüntü iyileştirme ile ilgili olarak literatüre bakıldığında da yine pek çok çalışma olduğu görülmektedir. Örneğin Kim (Kim ve diğerleri(1997)) ve arkadaşları yaptıkları çalışmada birinci türev ve yerel istatistiklere dayanan mamografi görüntüleri için uyarlanabilir bir görüntü iyileştirme yöntemi önermişlerdir. Bu yöntem üç aşamadan oluşmaktadır. Birinci aşamada mikrokalsifikasyon zannedilen artefektler kaldırılmıştır. İkinci aşamada, birinci türev operatörleri kullanılarak görüntü değişiklikleri hesaplanmaktadır. Son aşamada ise uyarlamalı ağırlıklı gradyan görüntüleri ile mamografi imgesinin önemli özellikleri geliştirilmektedir. Uyarlamalı olarak görüntü detaylarının güçlendirilebilmesi ve görüntü gürültülerinin bastırılabilmesi için görüntünün yerel istatistiğinden faydalanmışlardır. Amutha (Amutha ve diğerleri(2011)) ve arkadaşları ise çalışmalarında, meme kanserinin erken tanısı ve teşhisi için kontrast artırma ve gürültü giderme amacıyla bir algoritma sunmuşlardır. Bu çalışma modifiye matematiksel morfoloji ve biortogonal dalgacık analizine dayanmaktadır. Önerilen algoritmanın performans değerlendirmesi için Kontrast Artış İndeksi (CII) ve Kenar Koruma İndeksi (EPI) kullanılmıştır. Sonuçlar, önerilen algoritmanın diğer iyi bilinen algoritmalara kıyasla çok daha iyi görüntü kalitesi sağladığını göstermektedir. Moradmand ve arkadaşları (Moradmand ve diğerleri (2012)) dijital mamogramlarda görüntü iyileştirme amacıyla dört farklı algoritmanın karşılaştırıldığı bir çalışma sunmuşlardır. Bu çalışmada dalgacık tabanlı iyileştirme (8. Seviye Asimetrik Daubechies), Kontrast Sınırlı Adaptif Histogram Eşitleme (CLAHE), morfolojik operatörler ve keskin olmayan maskeleme karşılaştırılmış ve 114 klinik dijital mamografi üzerinde test edilmiştir. Deneysel sonuçlar, dalgacık dönüşümün daha etkili olabileceğini ve özellikle yoğun meme yapıları için BDT sisteminin genel olarak iyileştirilebileceğini düşündürmektedir. Akila (Akila, Jayashree, Vasuki(2015)) ve arkadaşları dolaylı kontrast geliştirme tekniklerinden histogram eşitleme(HE), kontrast sınırlı adaptif histogram eşitleme(CLAHE), Parlaklığı Koruyan İki Boyutlu Histogram Eşitleme(BBHE), Minimum Ortalama Parlaklık Hatası İki Boyutlu Histogram Eşitleme (MMBEBHE), Özyineli Ortalama-Ayrı Histogram Eşitleme (RMSHE) yöntemlerini düşük kontrastlı mamografi görüntülerine uygulayarak karşılaştırılmıştır. Hem CLAHE hem de RMSHE tekniklerinin, test görüntülerinde bulunan kitlelerde ve mikrokalsifikasyonlarda daha iyi iyileştirilme sağladığı görülmüştür. Ayrıca, RMSHE tekniğinin yapılan performans değerlendirme çalışmalarına göre düşük kontrastlı mamografi görüntüleri için en iyi kontrast arttırmayı sağladığı ve aynı zamanda en iyi parlaklık koruma özelliğini sunduğu görülmüştür. AlSalman (AlSalman(2019)) yaptığı çalışmada görüntü alanında fazla bilgi kaybı olmadan, frekans alanında dijital mamogram meme görüntülerini iyileştirmek için Butterworth bant geçiren filtrenin kullanılması için bir yaklaşım önermiştir. Değerlendirme aşamasında miniMIAS veritabanından on görüntü rastgele seçilmiş ve önerilen yaklaşıma dayalı olarak iyileştirilmiştir. Elde edilen sonuçlar Yapısal İçerik (SC), Ortalama Kare Hatası (MSE), Normalleştirilmiş Çapraz Korelasyon (NCC), Pik Sinyal Gürültü Oranı (dB cinsinden PSNR), Maksimum Fark (MD), Ortalama Fark (AD),ve Normalleştirilmiş Mutlak Hata (NAE) metrikleri açısından değerlendirilmiştir. Önerilen yaklaşımda kullanılan Butterworth bant geçiren filtrenin sonuçları ile medyan filtrenin sonuçları karşılaştırılmıştır. Karşılaştırma sonuçlarına

MEME KANSERLERİNİN TEŞHİSİNDE KULLANILAN GÖRÜNTÜ İŞLEME TEKNİKLERİYLE İLGİLİ BİR DERLEME ÇALIŞMASI

göre önerilen yaklaşımın aynı görüntülerde MSE, PSNR, NCC ve AD görüntü kalitesi metrikleri açısından medyan filtreye kıyasla en iyi sonuçları verdiği görülmüştür. Duan ve arkadaşları (Duan ve diğerleri (2019)) mamografi görüntülerini iyileştirmede kullanılan bazı yöntemlerin kenarları zayıflattığını veya ekstra gürültü eklediğini belirtmişler ve bu sorunu gidermek üzere yaptıkları çalışmada Laplacian piramidinde dinamik bir keskinliği azaltma maskesi kullanarak mamogram için çok ölçekli bir kontrast geliştirme önermişlerdir. Laplacian piramidi her ölçekte ince yapıyı korumak için kullanılırken, dinamik keskinliği azaltma maskesi ayrıntıları uyarlamalı olarak geliştirmek ve aynı anda gürültüyü bastırmak için kullanılmıştır. Önerilen yöntem görüntünün aşağı örneklenmesi, dinamik keskinliği azaltma maskesinin ağırlığını hesaplanması, her ölçek görüntüsü için dinamik keskinliği azaltma maskesinin kullanılması, Laplacian piramid görüntülerini elde etmek için her bir ölçek görüntüsü için geliştirme sonuçlarının çıkarılması ve son olarak geliştirilmiş görüntünün elde edilmesi olmak üzere beş aşamadan oluşmaktadır. Önerilen yöntemi kalitatif ve kantitatif olarak değerlendirmek için bir fantom ve üç klinik mamografi olgusu kullanılmıştır. Sonuçlar, önerilen yöntemin meme bezinin çok daha net ayrıntılarını sağlayabildiğini göstermiştir. Kantitatif olarak, bilgi entropisi ve tepe sinyal-gürültü oranı, en son teknoloji ile karşılaştırıldığında, en fazla 0.96 ve 3.89 oranında artırılmıştır. Önerilen yöntem, ayrıntıları uyarlamalı olarak arttırmak ve gürültüyü nispeten bastırmak için büyük bir potansiyele sahip olan bölgesel bir uyarlanabilir evrimin yardımıyla farklı bölge türlerinin iyileştirildiğini göstermiştir. Haindl ve Remeš (Haindl, Remeš (2019)) çalışmalarında mamogram iyileştirme için tam otomatik üç tamamlayıcı hedefli çıkış ile yeni hızlı bir yöntem önermişlerdir. Mamogram görüntüsünün iyileştirilmesi birkaç yüz dokusal istatistiğin verimli bir şekilde hesaplanması, onların alt dekokorelasyonları ve orijinal mamogramın ve dönüştürülmüş özelliklerinden en çok bilgi veren üç özelliğin ağırlıklı kombinasyonunun görselleştirilmesine dayanmaktadır. Algoritma rastgele alan tipi bir model kullansa da, verimli yinelemeli ve sayısal olarak sağlam model tahmini sayesinde bu model oldukça hızlıdır. Kanserli alanlar tipik olarak dokusal kusurlar olarak mamogram görüntülerinde kendini gösterebilmektedir. Yapılan çalışma, dokusal anormallikler olarak algılanan ilgi alanlarını ortaya çıkarmakta ve bu çalışma sayesinde iyileştirilmiş mamogramlar radyologların yanlış negatif değerlendirme oranlarını azaltmalarına yardımcı olabilmektedir. Algoritmanın hem mikro kalsifikasyonlar gibi küçük bulgular hem de daha büyük lezyonlar için iyi çalıştığı gösterilmiştir. Örneklerde, çevreleriyle aynı ortalama gri seviyeye sahip zar zor görünen anormalliklerin bile kullanılan yöntem sayesinde çok iyi vurgulandığı ve meme kanseri tespitinde büyük fayda sağlayabileceği görülebilmektedir.

5.2. Bölütleme Aşaması

Meme kanserinin tanısı ve teşhisinde bölütleme en önemli adımlardan birisi olarak karşımıza çıkmaktadır. Bölütleme genel olarak şüpheli bölgelerin diğer bölgelerden ayrılması olarak düşünülebilir. Bu amaçla yapılan ilk çalışmalardan biri Strauss (Strauss ve diğerleri (1992)) ve arkadaşları tarafından sayısal mamogramlarda mikrokalsifikasyonun tespiti ve bölütlenmesi için bilgisayar tabanlı yöntemlerin kullanılmasının incelendiği çalışmadır. Bilgisayar algılama sistemi otomatik olarak elde edilen r ve h parametrelerinin, iki filtrelenmiş görüntüye iki kez uygulandığı rhmaxima algoritmasına dayanmaktadır. Bölütleme işlemi, arka plan ve mikrokalsifikasyonlar için farklı işaretlerin kullanıldığı watershed dönüşümünden türetilen bir algoritma ile gerçekleştirilmiştir. Bir diğer çalışma Li (Li ve diğerleri (1993)) ve diğerlerinin gerçekleştirdiği çalışmadır. Bu çalışmada tümörün veya diğer şüpheli bölgelerin tanımlanması amacıyla dijital mamogramların bölütlenmesi için bir yöntem önerilmiştir. Bu yöntem bir görüntünün modellenmesi için gauss işlemi ile Markov Rasgele Alanını birleştirmektedir. Önerilen yöntem 13 mamografi üzerinde test edilmiş ve yeni tekniğin dijital mamogramlardaki tümörlerin teşhisinde önemli bir potansiyele sahip olduğu bildirilmiştir. Sahakyan ve Sarukhanyan (Sahakyan ve Sarukhanyan(2012)) çalışmalarında arka plan bölgesinden göğüs profil bölgesini ayırmak ve sayısallaştırma gürültüsünü gidermek için ileri kenar algılama ve bölge bölütlemesi amacıyla morfolojik ön işleme algoritmasını kullanmışlardır. Önerilen algoritma farklı meme yoğunluğuna sahip mamogramlarda test edilmiş ve genel olarak meme bölgesi tespiti amacıyla kalite ölçeği ortalama değeri 0.95 olarak elde edilmiştir, bu değer algoritmanın farklı yoğunluklarda bile son derece sağlam olduğunu göstermektedir. Isa ve Siong (Isa ve Siong (2012)) mamogramlarda kitle segmentasyonu ve tespiti ile ilgili çalışmalarında ilk olarak mamogramı, meme

MEME KANSERLERİNİN TEŞHİSİNDE KULLANILAN GÖRÜNTÜ İŞLEME TEKNİKLERİYLE İLGİLİ BİR DERLEME ÇALIŞMASI

bölgesi ve meme olmayan bölge olarak ikiye ayırmışlar daha sonra görüntü iyileştirme amacıyla mamogramlardaki doku yapısının kontrastını iyileştirmişlerdir. Son olarak mamogramlarda kitle tespiti ve bölünmesi için lokal istatistiksel doku analizine dayanan sınırlı bölge büyütme uygulamışlardır. Önerdikleri sistemi MIAS veri tabanında bulunan 322 görüntü üzerinde test etmişlerdir. Elde ettikleri veriler önerilen tekniğin %94.59 hassasiyetle ve resim başına 3.9 yanlış pozitif sayısı ile kitleleri tespit ettiğini göstermektedir. Cheng (Cheng ve diğerleri (2006)) ve arkadaşları yaptıkları kapsamlı çalışmada kitle tespiti ve sınıflandırma yöntemlerini incelemişler ve bu yöntemlerin avantaj ve dezavantajlarını karşılaştırmışlardır. Gulsrud (Gulsrud, Engan, Hanstveit(2005)) ve diğerleri dijital mamografide tespit edilen kitlenin bölütlenmesi için gri ölçekli matematiksel morfolojiye dayanan bir yöntem sunmuşlardır. Ön işleme aşamasında, lokal histogram tekniğine dayalı görüntü iyileştirmesi sonrasında morfolojik düzleştirme işlemi uygulamışlardır. Bölütlenmiş bölgelerdeki yumuşatılmış görüntünün gradyanına watershed dönüşümü uygulamışlardır. Önerilen yöntem MIAS veritabanında bulunan mamogramlar üzerinde test edilmiştir. Deneysel sonuçlar, yöntemin kitle yapılarının konturunu çıkarmada faydalı olacağını göstermektedir. Das ve Das (Das ve Das (2019)) çalışmalarında, dağınık mamogramlarda kitleleri segmentlere ayırmak için yeni bir çekirdek bazlı FCM kümeleme yaklaşımı kullanmışlardır. Kitleler ve normal glandüler dokular arasındaki marjinal parametrik farklılıkları taşıyan önemli özelliklerin seçilmesi, önerilen segmentasyon yaklaşımı için bir zorluktur. Birçok araştırmada, ilgilenilen bölgeyi kümelemek ve segmentlere ayırmak için mekansal yoğunluk varyasyonundan faydalanılmaktadır. Bununla birlikte, bazen mamogramdaki hiper yoğun glandüler dokular, kan damarları ve pektoral kaslar, kitleler ile benzer bir parlaklık oluşturur ve yanlış sonuçlara neden olurlar. Yapılan çalışmada, yoğun glandüler dokuların doku dağılımının rastgele ve kaba olduğu bulunmuş ve bu nedenle bu bölgeler doku temelli değerlendirmeyi ele alarak kolayca tanımlanabilmiş ve bastırılabilmiştir. Çalışmada önerilen yöntem, yerel yoğunluk varyasyonunu ve bulanık mantık tabanlı kümeleme teorisinin özelliklerini kullanarak, mamografik görüntülerin belirsizliği ve belirsizliğinin doğasında bulunan doğayı da çözebilmektedir. Ayrıca, önerilen bölütleme yaklaşımını başlatmak için manuel bir müdahaleye gerek yoktur. Bu çalışma aynı zamanda önerilen yaklaşımın performansını FCM yöntemi ve k-Means yöntemi gibi diğer kümelemeye dayalı segmentasyon teknikleriyle değerlendirmek için karşılaştırmalı bir değerlendirme sunmuştur. Shrivastava ve Bharti (Shrivastava ve Bharti (2020)), meme tümörünü bir ilgi alanı (ROI) içinde segmentlere ayırmak için tasarlanmış koşullu bir üretken çekirdekli ağ (cGAN) önermişlerdir. Üretken ağ, tümör alanını tanımayı ve ana hatları çizen ikili maskeleri oluşturmayı öğrenirken çekirdekli ağ, doğrulama verileri ve sentetik segmentasyonlar arasında ayırım yapmayı öğrenerek, üretken ağın mümkün olduğunca gerçekçi ikili maskeler oluşturmasını zorunlu kılmaktadır. Yapılan çalışma, INbreast ve özel bir şirket içi veri seti üzerinde yapılan çeşitli segmentasyon deneyleri ile test edilmiştir. Önerilen segmentasyon modeli, tümör içeren görüntüler ve sağlıklı dokuların yer aldığı görüntüler üzerinde, sırasıyla % 94 ve % 87 oranında Dice katsayısı ve Birleşme Üzerinde Kesişim (IoU) elde etmiştir. Ek olarak, üretilen maskeleri düzensiz, lobüler, oval ve yuvarlak olmak üzere dört tümör şekline göre sınıflandırmak için Konvolüsyonel Sinir Ağına (CNN) dayanan bir şekil tanımlayıcı önerilmiştir. Senthilkumar ve Umamaheswari (Senthilkumar ve Umamaheswari (2015)) meme kanserinin etkili tespiti için yeni Eşitsiz Aralıklanan Hızlı Fourier Dönüşümü (USFFT) ve 2D Hızlı Ayrık Curvelet Dönüşümünü (FDCT) birleştirerek hem gürültüyü azaltmışlar hem de mamogram görüntüsünü iyileştirmişlerdir. Çalışmalarını test etmek amacıyla Mini-MIAS veri tabanından aldıkları görüntü için Ortalama Kare Hatası (MSE) ve Tepe Sinyal / Gürültü Oranı (PSNR) hesaplamışlardır. Yapılan çalışma, Modifiye Yerel Aralık Modifikasyonu (MLRM) ve Gaussian Laplacian (LoG) kenar algılama yöntemini birleştiren yeni bir segmentasyon yöntemi içermektedir. Sonuçlar ROC grafiği kullanılarak analiz edilmiş ve bulunan algılama doğruluğu, mevcut yöntemlere kıyasla % 99 olarak bulunmuştur.

5.3. Öznitelik Çıkarımı Aşaması

Meme kanserinin tespitine yönelik bir diğer aşama öznitelik çıkarımıdır. Öznitelik çıkarımı tüm görüntünün kullanılması yerine yalnızca görüntünün belirli özelliklerinin işleme tabi tutulmasıdır. Doku analizi ile ilgili öznitelik çıkarımına dair yaklaşımlar dört başlıkta toplanabilir. Bunlar; yapısal, istatistiksel, model tabanlı ve dönüşüm yöntemlerine dayanan yaklaşımlardır. Örneğin Haralick gri ton

MEME KANSERLERİNİN TEŞHİSİNDE KULLANILAN GÖRÜNTÜ İŞLEME TEKNİKLERİYLE İLGİLİ BİR DERLEME ÇALIŞMASI

uzaysal bağımlılıklarına dayanan bazı kolay hesaplanabilir dokusal özellikler ve üç farklı görüntü verisinin tanımlanmasına ait uygulamaları gerçekleştirdiği bir çalışma yapmıştır. Elde ettiği sonuçlar bu yöntemin, kolayca hesaplanabilen dokusal özniteliklerin çok çeşitli görüntü sınıflandırma uygulamaları için genel bir uygulanabilirliğe sahip olduğunu göstermektedir(Haralick, Shanmugam, Dinstein (1973)). Lahmiri ve Boukadoum (Lahmiri ve Boukadoum (2011)) kanser görüntülerini sınıflandırmak için ayrı dalgacık dönüşümü ve Gabor filtresi kombinasyonundan elde edilen istatistiksel özellikleri kullanan bir sistem önermişlerdir. Bu sistem sınıflandırıcı olarak destek vektör makinelerini kullanmaktadır. Deneysel sonuçlar, böyle karma bir işleme modelinin, DWT ve Gabor filtre bankalarının tek başına kullanımıyla karşılaştırıldığında daha düşük standart sapma ve daha yüksek doğruluk sağladığını göstermektedir. Al-Shamlan ve El-Zaart(Al-Shamlan ve El Zaart(2010)) çalışmalarında, mamografi görüntüsünde daha fazla anlam ifade eden 23 önemli öznitelik çıkartmışlardır. Öznitelikler seçilirken mamografi görüntü alanındaki uzman kişilerle yapılan birçok röportaj ve bu alanda yayınlanmış birçok makaleyi temel almışlardır. Kullanılan bu öznitelikleri 80 mamografiye uygulamışlar ve çok iyi sonuçlar elde ettiklerini bildirmişlerdir. Saraswathi (Saraswathi, Dharani, Srinivasan (2016)) ve diğerleri dijital mamografinin analizi için verimli bir öznitelik çıkarma yöntemi uygulamışlardır. Başlangıçta, her mamogram görüntüsü ayrı ayrı dalgacık ve eğricik dönüşümleri kullanılarak çok seviyeli ayrışma ve sarmalama işlemine tabi tutulmuştur. Ardından, her ayrışma seviyesine ait katsayılar elde edilmiş ve eğricik katsayıları destek vektör makinesine verilmiştir. Performans sınıflandırma doğruluk oranı kullanılarak ölçülmüştür. Deneysel sonuçlar, eğriciğin dalgacıktan daha iyi bir performans sağladığını göstermektedir. Ayrıca elde edilen eğricik katsayıları sınıflandırılmayacak kadar fazla olduğu için zaman karmaşıklığını azaltmak ve belirgin özellikleri seçmek amacıyla bir Parçacık Sürü Optimizasyon (PSO) algoritması kullanmışlardır. Öznitelik çıkarımı ile elde özniteliklerin boyutu bazen çok büyük olabilmektedir. Bu gibi durumlarda özniteliklerin azaltılması ve seçimi amacıyla farklı algoritmalar ve yöntemler geliştirilmiştir. Örneğin Liu ve Tang (Liu ve Tang (2014)) meme kitlelerinin sınıflandırılmasını 408 kötü huylu ve 418 iyi huylu kitlesi olan 826 ilgi bölgesi elde edilmiş veri setiyle araştırmış ve sunmuşlardır. Bölütleme amacıyla seviye belirleme yöntemiyle mekansal FCM kümeleme yöntemini birleştirmişlerdir. Bölütlemeyen sonra, her kitle için 12 geometrik öznitelik ve 19 doku özelliği olmak üzere 31 öznitelik elde etmişlerdir. F-skoru, Relief, SVM-RFE, SVM-RFE (mRMR) ve önerilen SRN de dâhil olmak üzere birçok öznitelik seçimi yöntemini tartışmışlardır ve önerdikleri öznitelik seçim yönteminin, daha az seçilmiş özniteliklere sahip diğer seçim yöntemlerinden daha iyi performans gösterdiğini belirtmişlerdir. Nandi (Nandi ve diğerleri (2006)) ve arkadaşları 57 kitle içeren mamogram görüntüsüne ait kenar keskinliği, şekli ve dokusu ile ilgili 22 özelliğin çıkarıldığı bir veri seti kullanmışlardır. Çalışmalarındaki yenilik, genetik programlamanın (GP) uyarlanması ve uygulanmasıdır. GP sınıflandırıcısının kullandığı öznitelik havuzunu hassaslaştırmak için Student'ın t-testi, Kolmogorov-Smirnov Testi ve Kullback-Leibler ayrıcısı olmak üzere üç istatistiksel ölçüt içeren beş öznitelik seçimi yöntemi kullanmışlardır. Elde edilen doğruluk sonuçlarının, genel olarak eğitim için % 99.5 ve üzeri ve test için % 98 ve üzeri olduğunu belirtmişlerdir. Nugroho (Nugroho ve diğerleri(2014)) ve arkadaşları ise yaptıkları çalışmada elde ettikleri mamogramlar için histogram ve Gri Seviye Eş-oluşum Matrisini(GLCM) ile ilgili toplam 12 öznitelik çıkarmışlardır. Çalışmalarında özniteliklerin% 50'sini azaltan Korelasyona Dayalı Özellik Seçimi (CFS) kullanılmışlardır. Seçilmiş bu özellikler çok katmanlı algılayıcı algoritması ile sınıflandırılmıştır. Özel Onkoloji Kliniği Kotabaru Yogyakarta'dan alınan 40 dijital mamografi verisi için elde edilen deneysel sonuçlar % 91.66 doğruluk oranına ulaşıldığını göstermektedir. Vijayarajeswari (Vijayarajeswari ve diğerleri (2019)) ve arkadaşları Mini-MIAS veritabanından aldıkları 95 mamogram görüntüsü üzerinden özellik çıkarımı amacıyla Hough dönüşümünden faydalanmıştır. Hough dönüşümü iki boyutlu bir dönüşüm olup görüntüdeki belirli şeklin özelliğini izole etmek için kullanılır. Çıkarılan özellikler SVM kullanılarak iyi huylu ve kötü huylu olarak sınıflandırılmıştır. SVM ile sınıflandırılan 95 mamogram görüntüsü için %94 oranında doğruluk oranı elde ettiklerini bildirmişlerdir.

5.4. Sınıflandırma Aşaması

Son aşama sınıflandırma ve karar aşamasıdır. Meme kanserinin sınıflandırılmasında genel olarak kanserli veya kanserli değil ve iyi huylu veya kötü huylu şeklinde iki sonuç elde edildiği

MEME KANSERLERİNİN TEŞHİSİNDE KULLANILAN GÖRÜNTÜ İŞLEME TEKNİKLERİYLE İLGİLİ BİR DERLEME ÇALIŞMASI

gözlemlenmektedir. Sınıflandırıcı olarak pek çok yöntem kullanılmaktadır. Örneğin Rabidas (Rabidas, Midya, Chakraborty (2018)) ve arkadaşları mamografi kitlelerinin iyi huylu ve kötü huylu olarak sınıflandırılmasında Yapısal Komşuluk Benzerliğini (NSS) kullanan iki yeni öznelik çıkarım yöntemi önermişlerdir. İyi huylu ve kötü huylu kitlelerdeki piksellerin gri seviye dağılımı arasındaki farktan yararlanan yöntem kitlelerin komşu bölgelerle arasındaki benzerliği kullanmaktadır. NSS-I ve NSS-II, olarak adlandırılan bu iki yeni yöntem farklı ölçeklerde global benzerlik yakalamaktadır. Yöntemin performansı, mini-MIAS ve DDSM veri tabanlarından alınan görüntülerle Fisher lineer ayırıcı analizi ile 10 kat çapraz doğrulama yöntemi kullanılarak değerlendirilmiştir. Yazarlar Mini-MIAS veritabanı için % 94.57, DDSM veritabanı için % 85.42 doğruluk elde ettiklerini belirtmişlerdir. Elman ve Kadah (Elman ve Kadah(2015)) kitleler ile normal göğüs dokusu arasındaki farkı belirlemek için bir CAD sistemi önermişlerdir. İlk önce DDSM veri tabanından alınan görüntüler önışlemeden geçirilmiş ve mamogram görüntüleri iyileştirilmiştir, ardından kitleler ve normal göğüs dokusu içeren 100 ilgi bölgesi (ROI) çıkartılmıştır. Elde edilen ilgi bölgeleri için 59 dokusal ve istatistiksel öznelik çıkartmışlardır. Ardından öznelik seçimi için Ardışık İleri Seçme ve Ardışık Dalgalı İleri Seçme yöntemlerini kullanmışlardır. Son olarak, sınıflandırma için K-En Yakın Komşuluk (K-EK) algoritması, Doğrusal Ayrımcılık Analizi (LAA), Karesel Ayrımcılık Analizi (QDA) ve Destek Vektör Makinalarını (DVM) kullanmışlardır. Şerifović-Trbalić (Şerifović-Trbalić ve diğerleri (2014)) ve arkadaşları, mamografi görüntülerinden kitle saptama ve sınıflandırma için yeni bir bilgisayar destekli tanı sistemi önerilmiştir. Önerilen sistem tamamen otonom bir sistemdir. Dijital mamogramlardan ilgi alanlarının çıkartılması için görüntü önışleme ve Otsu N eşikleme algoritması kullanılmıştır. Daha sonra ilgi alanlarından farklı düzen ve iterasyonlarda bir grup 32 Zernike momenti çıkarılmıştır. Bu momentler esnek geri yayılım nöral ağ sınıflandırıcıya uygulanmıştır. Deneysel sonuçlar, önerilen tekniğin, dijital mamografide kötü huylu ve iyi huylu kitle tipindeki anormalliklerin etkili bir şekilde sınıflandırılabilirdiğini göstermektedir. Jen ve Yu (Jen ve Yu(2015)) çalışmalarında birinci dereceden istatistiksel yoğunluklar ve gradyanlara dayalı öznelikler çıkararak yeni anormallik saptama sınıflandırıcısına (ADC) gönderen ve anormal mamografileri tespit eden bir yöntem önermişlerdir. Çalışmalarında, öncelikle daha doğru göğüs segmentasyonu için küresel eşitleme dönüşümü, gürültü giderme, ikilik resme dönüştürme, meme oryantasyonu tayini ve pektoral kası gizleme gibi önışleme adımlarını araştırmışlardır. Öznelik ağırlıklarının belirlenmesi için Temel Bileşen Analizini (TBA) kullanmışlardır. Deneysel sonuçlar, anormal mamografi tespiti için ADC algoritmasının öznelik ağırlık ayarlamaları ile uygulanmasının, ilgili iki veri kümesinde % 88 ve % 86'lık belirgin hassaslık verdiğini göstermektedir. Setiawan (Setiawan ve diğerleri(2015)) ve arkadaşları çalışmalarında, dokusal özellik çıkarma yöntemi olarak LAWS'ın Doku Enerji Ölçümü'nü (LAWS) kullanmışlardır. Mamogramların normal anormal ve iyi huylu- kötü huylu görüntüler için sınıflandırıcı olarak Yapay Sinir Ağı (YSA) kullanmışlardır. Test verileri için MIAS'tan alınan mamogram görüntülerini kullanmışlardır. Yaptıkları çalışma LAWS ile özellik çıkarımının GLCM'ye oranla daha iyi doğruluk sağladığını göstermektedir. LAWS normal anormal sınıflandırmada % 93,90 doğruluk, iyi huylu- kötü huylu sınıflandırma için% 83,30 doğruluk sağlarken, GLCM normal-anormal sınıflandırmada % 72,20 doğruluk, iyi huylu- kötü huylu sınıflandırmada ise % 53.06 doğruluk kaydetmiştir. Nithya ve Santi (Nithya ve Santi (2011)) mamogramların normal ve kanserli olarak sınıflandırıldığı bir çalışma yürütmüşlerdir. Çalışmalarında mamografi görüntülerinden GLCM özneliklerinin çıkartılması ve bu özneliklerin sınıflandırılmasını araştırmışlardır. Normal ve kanser sınıflandırması için maksimum doğruluk oranını %96 olarak elde etmişlerdir. Eltoukhy (Eltoukhy, Faye, Samir (2012)) ve diğerleri, dijital mamografi görüntülerinden meme kanseri tanısı için bir yöntem önermişlerdir. Dalgacık ve Eğriciğin çok çözünürlüklü gösterimini mamogram görüntülerinin uzun bir katsayı vektörüne dönüştürülmesi için kullanılmışlardır. Dalgacık ve Eğricik katsayılarından oluşan satırların görüntü sayısını sütunlarında katsayı sayısını belirttiği bir matris oluşturmuşlardır. İstatistiksel t-testi yöntemine dayalı olarak bir öznelik çıkarma yöntemi geliştirmişlerdir. Geliştirilen yöntem, sınıfları ayırt etme kabiliyetine göre özneliklerin (sütunların) sıralamasını yapmaktır. Sınıflandırma yapmak için Destek vektör makinesi (SVM) kullanmışlardır. Sınıflandırma hem normal ve anormal dokular arasında hem de iyi huylu-kötü huylu tümörleri ayırmak için kullanılmıştır. Elde ettikleri sınıflandırma doğruluk oranları, önerilen yöntemin meme kanserinin başarılı bir şekilde saptanmasına katkıda bulunabileceğini göstermektedir.

6. SONUÇ

Bu bölümde dünyada ve ülkemizde kadınlarda görülen en önemli kanser türlerinden biri olan meme kanseri konusu ele alınmıştır. Bu amaçla ilk olarak meme anatomisi anlatılmış ardından meme kanseri ve bu kanser ile ilgili bazı istatistiki bilgiler verilmiştir. Bölümde ayrıca meme kanserinin erken teşhisinde uzmana yardımcı olan BDT sistemleri konusunda bir literatür taraması yapılmış ve elde edilen çalışmalar BDT sistemlerinin teşhis yapılırken mamogramlar üzerinde uygulanan aşamalarına göre sınıflandırılarak sunulmuştur. Bu çalışma ile meme kanseri için bir BDT sistemi geliştirmek isteyen araştırmacılara kendi BDT sistemlerinin her bir aşamasında hangi yöntemi kullanmaları konusunda fikir vermesi hedeflenmiştir.

7. KAYNAKÇA

AlSalman Hussain (2019). Enhancing Digital Mammogram Images using Bandpass Filters in Frequency Domain. *IJCSNS International Journal of Computer Science and Network Security*, 19(11), 107-113

Al-Shamlan H. and El-Zaart A. (2010). Feature extraction values for breast cancer mammography images. 2010 International Conference on Bioinformatics and Biomedical Technology, Chengdu, 335-340. doi: 10.1109/ICBBT.2010.5478947

Akila K., Jayashree L.S., Vasuki A. (2015). Mammographic Image Enhancement Using Indirect Contrast Enhancement Techniques – A Comparative Study, *Procedia Computer Science*, 47, 255-261. doi:10.1016/j.procs.2015.03.205.

American College of Radiology (ACR)(1998). *Illustrated Breast Imaging Reporting and Data System (BI-RADS)*. Third Edition. Reston (VA)

Amutha S., Ramesh Babu D. R., Ravi Shankar M. and Harish Kumar N. (2011). Mammographic image enhancement using modified mathematical morphology and Bi-orthogonal wavelet. 2011 IEEE International Symposium on IT in Medicine and Education, Cuangzhou, 548-553. doi: 10.1109/ITIME.2011.6130897

Ateş S. (2014). Kadınların Kendi Kendine Meme Muayenesi Uygulamalarının Değerlendirilmesi (Yüksek Lisans Tezi). Erişim adresi: <https://tez.yok.gov.tr/UlusalTezMerkezi/tezSorguSonucYeni.jsp>

Cheng H.D., Shi X.J., Min R., Hu L.M., Cai X.P., Du H.N. (2006). Approaches for automated detection and classification of masses in mammograms. *Pattern Recognition*, 39(4), 646-668. doi.org/10.1016/j.patcog.2005.07.006.

Ciatto S., Ambrogetti D., Collini G., Cruciani A., Ercolini E., Risso G., Rosselli Del Turco M.(2006). Computer-aided detection (CAD) of cancers detected on double reading by one reader only. *The Breast*, 15(4), 528-532. doi:10.1016/j.breast.2005.08.035.

Das Poulomi, Das Arpita (2019). A fast and automated segmentation method for detection of masses using folded kernel based fuzzy c-means clustering algorithm. *Applied Soft Computing*, 85, 105775. doi:10.1016/j.asoc.2019.105775.

Dheeba J., Albert Singh N., Tamil Selvi S. (2014). Computer-aided detection of breast cancer on mammograms: A swarm intelligence optimized wavelet neural network approach. *Journal of Biomedical Informatics*, 49, 45-52. doi:10.1016/j.jbi.2014.01.010.

Duan X. et al. (2019). A Multiscale Contrast Enhancement for Mammogram Using Dynamic Unsharp Masking in Laplacian Pyramid. *IEEE Transactions on Radiation and Plasma Medical Sciences*, 3(5), 557-564. doi: 10.1109/TRPMS.2018.2876873

Elmanna M. E. and Kadah Y. M. (2015). Implementation of practical computer aided diagnosis system for classification of masses in digital mammograms. 2015 International Conference

MEME KANSERLERİNİN TEŞHİSİNDE KULLANILAN GÖRÜNTÜ İŞLEME TEKNİKLERİYLE İLGİLİ BİR DERLEME ÇALIŞMASI

on Computing, Control, Networking, Electronics and Embedded Systems Engineering (ICCNEEE), Khartoum, 336-341. doi: 10.1109/ICCNEEE.2015.7381387

Eltoukhy, M.M., Faye, I. ve Samir, B.B.(2012). A statistical based feature extraction method for breast cancer diagnosis in digital mammogram using multi resolution representation. Computers in Biology and Medicine, 42, 123-128. DOI:[10.1016/j.combiomed.2011.10.016](https://doi.org/10.1016/j.combiomed.2011.10.016)

Gedik N.(2013). Bilgisayar Destekli Teşhis Sistemi İle Mamografi Görüntülerinin Sınıflandırılması (Doktora Tezi). Erişim adresi: <https://tez.yok.gov.tr/UlusalTezMerkezi/tezSorguSonucYeni.jsp>

Gowri D. S. and Amudha T., (2014). A Review on Mammogram Image Enhancement Techniques for Breast Cancer Detection. 2014 International Conference on Intelligent Computing Applications, Coimbatore, 47-51.doi: 10.1109/ICICA.2014.19

Gulsrud T. O., Engan K. and Hanstveit T. (2005). Watershed segmentation of detected masses in digital mammograms.2005 IEEE Engineering in Medicine and Biology 27th Annual Conference, Shanghai, 3304-3307. DOI:[10.1109/IEMBS.2005.1617183](https://doi.org/10.1109/IEMBS.2005.1617183)

Güllüoğlu B. M.(2008). Meme hastalıklarına yaklaşım:“Meme kanseri için risk değerlendirmesi ve tarama stratejileri”, Türk Aile Hek Derg 2008; 12(1),9-1.

Haindl M., Remeš, V. (2019). Pseudocolor enhancement of mammogram texture abnormalities. Machine Vision and Applications 30, 785–794 .doi:[10.1007/s00138-019-01028-6](https://doi.org/10.1007/s00138-019-01028-6)

Hamed G., Marey M., Amin S. E. and Tolba M. F. (2018). A Proposed Model for Denoising Breast Mammogram Images. 2018 13th International Conference on Computer Engineering and Systems (ICCES), Cairo, Egypt, 652-657. doi: 10.1109/ICCES.2018.8639307

Haralick R. M., Shanmugam K. and Dinstein I. (1973) .Textural Features for Image Classification. IEEE Transactions on Systems, Man, and Cybernetics, SMC-3(6), 610-621. doi: 10.1109/TSMC.1973.4309314

Isa Nor Ashidi Mat and Siong Ting Shyue (2012). Automatic segmentation and detection of mass in digital mammograms. In Proceedings of the 11th international conference on Telecommunications and Informatics, Proceedings of the 11th international conference on Signal Processing (SITE'12). World Scientific and Engineering Academy and Society (WSEAS), Stevens Point, Wisconsin, USA, 143–146.

Jen Chun-Chu, Yu Shyr-Shen,(2015). Automatic detection of abnormal mammograms in mammographic images. Expert Systems with Applications,42 (6), 3048-305. doi:10.1016/j.eswa.2014.11.061.

Kim Jong Kook, Park Jeong Mi, Song Koun Sik and Park Hyun Wook (1997). Adaptive mammographic image enhancement using first derivative and local statistics.IEEE Transactions on Medical Imaging,16(5), 495-502.doi: 10.1109/42.640739

Kumar M., Thakkar V. M., Bhadauria H. S. and Kumar I. (2016). Mammogram's denoising in spatial and frequency domain. 2016 2nd International Conference on Next Generation Computing Technologies (NGCT), Dehradun, 654-659.doi: 10.1109/NGCT.2016.7877493

Kurt B.(2014). Mamografi Görüntülerinin Yorumlanmasıyla Bilgisayarlı Teşhis Sisteminin Tasarımı (Doktora Tezi). Erişim adresi: <https://tez.yok.gov.tr/UlusalTezMerkezi/tezSorguSonucYeni.jsp>

Lahmiri S. and Boukadoum M. (2011). Hybrid discrete wavelet transform and Gabor filter banks processing for mammogram features extraction. 2011 IEEE 9th International New Circuits and systems conference.Bordeaux, 53-56.doi: 10.1109/NEWCAS.2011.5981217

Li Huai-Dong, Kallergi M., Clarke L. P., Qian Wei and Clark R. A. (1993). Markov random field model for mammogram segmentation. Proceedings of the 15th Annual International Conference of the IEEE Engineering in Medicine and Biology Societ, 54-55.

MEME KANSERLERİNİN TEŞHİSİNDE KULLANILAN GÖRÜNTÜ İŞLEME TEKNİKLERİYLE İLGİLİ BİR DERLEME ÇALIŞMASI

Liu X. and Tang J. (2014) .Mass Classification in Mammograms Using Selected Geometry and Texture Features, and a New SVM-Based Feature Selection Method. *IEEE Systems Journal*, 8(3), 910-920. doi: 10.1109/JSYST.2013.2286539

Mencattini A., Salmeri M., Lojacono R., Frigerio M. and Caselli F. (2008). Mammographic Images Enhancement and Denoising for Breast Cancer Detection Using Dyadic Wavelet Processing. *IEEE Transactions on Instrumentation and Measurement*, 57(7), 1422-1430, doi:10.1109/TIM.2007.915470

Moradmand H., Setayeshi S., Karimian A. R., Sirous M., & Akbari M. E. (2012). Comparing the Performance of Image Enhancement Methods to Detect Microcalcification Clusters in Digital Mammography. *Iranian Journal of Cancer Prevention*, 5(2), 61–68.

Moreira Inês C., Amaral Igor, Domingues Inês, Cardoso António, Cardoso, Jaime S. Cardoso Maria João (2012). INBREAST, *Academic Radiology*,19(2),236-248.doi :10.1016/j.acra.2011.09.014.

Nandi R. J., Nandi A. K., Rangayyan R. and Scutt D. (2006). Genetic Programming and Feature Selection for Classification of Breast Masses in Mammograms. 2006 International Conference of the IEEE Engineering in Medicine and Biology Society, New York, NY, 3021-3024. DOI:[10.1109/IEMBS.2006.260460](https://doi.org/10.1109/IEMBS.2006.260460)

Nithya R., Santhi B. (2011). Classification Of Normal And Abnormal Patterns In Digital Mammogram for Diagnosis of Breast Cancer. *International Journal Of Computer Application*,28(6).

Nugroho H. A., Faisal N., Soesanti I. and Choridah L. (2014). Identification of malignant masses on digital mammogram images based on texture feature and correlation based feature selection. 2014 6th International Conference on Information Technology and Electrical Engineering (ICITEE), Yogyakarta,1-6.doi: 10.1109/ICITEED.2014.7007907

Özmen V. (2013). Türkiye'de Meme Kanseri, Türkiye Meme Hastalıkları Dernekleri Federasyonu (TMHDF),Ulusal Meme Kanseri Veritabanı (UMKVT) Verilerinin Analizi, Türkiye Klinikleri J Gen Surg-Special Topics. 6(2),1-6.

Özmen V., Cantürk Z., Çelik V., Güler N., Kapkaç M., Koyuncu A., Müslümanoğlu M., Utkan Z. (2012). Meme Hastalıkları Kitabı. Güneş Tıp Kitabevleri.

Rabidas R., Midya A. and Chakraborty J. (2018). Neighborhood Structural Similarity Mapping for the Classification of Masses in Mammograms. *IEEE Journal of Biomedical and Health Informatics*,22(3), 826-834. doi: 10.1109/JBHI.2017.2715021

Sahakyan A., & Sarukhanyan H. (2012). Segmentation of the Breast Region in Digital Mammograms and Detection of Masses. DOI:[10.14569/IJACSA.2012.030218](https://doi.org/10.14569/IJACSA.2012.030218)

Saraswathi D., Dharani D. and Srinivasan E. (2016). An efficient feature extraction technique for breast cancer diagnosis using curvelet transform and swarm intelligence. 2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), Chennai, 441-445. doi: 10.1109/WiSPNET.2016.7566172

Senthilkumar B., & Umamaheswari, G. (2015). Breast cancer detection using combined curvelet based enhancement and a novel segmentation methods. *Biomed Pap Med Fac Univ Palacky Olomouc Czech Repub.*, 159(1), 083-086. doi: 10.5507/bp.2013.097.

Setiawan A.S., Elysia Wesley J., & Purnama, Y. (2015). Mammogram Classification using Law's Texture Energy Measure and Neural Networks. DOI:[10.1016/j.procs.2015.07.341](https://doi.org/10.1016/j.procs.2015.07.341)

Shrivastava Neeraj & Bharti Jyoti (2020). Breast Tumor Detection in Digital Mammogram Based on Efficient Seed Region Growing Segmentation, *IETE Journal of Research*. DOI: [10.1080/03772063.2019.1710583](https://doi.org/10.1080/03772063.2019.1710583)

Strauss A. , Sebbar A., Désarnaud S., Mouillard P. and Le Gal M. (1992). Automatic detection and segmentation of microcalcifications on digitized mammograms. 1992 14th Annual

MEME KANSERLERİNİN TEŞHİSİNDE KULLANILAN GÖRÜNTÜ İŞLEME TEKNİKLERİYLE İLGİLİ BİR DERLEME ÇALIŞMASI

International Conference of the IEEE Engineering in Medicine and Biology Society, Paris,1938-1939.doi: 10.1109/IEMBS.1992.5762109

Šerifović-Trbalić A., Trbalić A., Demirović D., Prljača N. and Cattin P. C. (2014). Classification of benign and malignant masses in breast mammograms. 2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, 228-233. doi: 10.1109/MIPRO.2014.6859566

T.C. Sağlık Bakanlığı, Halk Sağlığı Genel Müdürlüğü. Erişim adresi: <https://hsgm.saglik.gov.tr/tr/kanser-anasayfa>

Toz G.(2018). Görüntü İşleme Teknikleriyle Meme Kanserinin Teşhisi (Doktora Tezi). Erişim adresi: <https://tez.yok.gov.tr/UlusalTezMerkezi/tezSorguSonucYeni.jsp>

Vijayarajeswari R., Parthasarathy P., Vivekanandan S., Alavudeen Basha A. (2019). Classification of mammogram for early detection of breast cancer using SVM classifier and Hough transform, Measurement, 146,Pages 800-805. [Doi:10.1016/j.measurement.2019.05.083](https://doi.org/10.1016/j.measurement.2019.05.083).

Vijikala V. and Dhas D. A. S. (2016).Identification of most preferential denoising method for mammogram images. 2016 Conference on Emerging Devices and Smart Systems (ICEDSS), Namakkal,173-179.doi: 10.1109/ICEDSS.2016.7587786

BULUT SİSTEMLERİNE YAPILAN DAĞITILMIS HİZMET REDDİ SALDIRILARI VE ALINABİLECEK ÖNLEMLER

Merve Yıldırım *

**Erzurum Teknik Üniversitesi, Yakutiye, Erzurum*

merve.yildirim@erzurum.edu.tr

** Sorumlu Yazar*

1. GİRİŞ

Bulut bilişim günümüzde sağladığı fırsatlar ve yönetsel kolaylıklar açısından organizasyonların en fazla kullandığı teknolojilerden bir tanesidir. Ancak, mevcut faydalarının yanı sıra ne yazık ki güvenlik zafiyetlerini de beraberinde getirmektedir. Bulut sistemlerine yapılacak olası saldırılar, organizasyonlar üzerinde yıkıcı etkilere sebep olabilir. Güvenlik saldırılarının niteliği de sürekli gelişmekte olduğundan yeni savunma mekanizmalarının geliştirilmesi zorunluluğu doğmuştur. Bu sebeple bilgi güvenliğinin temelini oluşturan gizlilik ve bütünlük kavramlarının sağlanması ve kurumların hizmet sürekliliğinin devam edebilmesi için bulut sistemlerini tehdit eden tüm kötü niyetli saldırılara karşı yeni güvenlik mekanizmaları geliştirilmektedir. Organizasyonların çoğu bulut sistemlerini kullanırken hala anti virüs programları ve güvenlik duvarları gibi geleneksel güvenlik önlemleriyle bilgi gizliliğini sağlamaya çalışsa da, bu yöntemler çoğu zaman yeni saldırı yöntemleriyle baş edebilecek kapasitede değildir. Bu çeşitli ve savunması zor saldırı yöntemleri kurumlara iş akışlarını engellemekten prestijlerini sarsmaya kadar pek çok zarar vermektedir. Bahsedilen saldırı yöntemlerinden en sık rastlanılanlarından ve en zararlı olanlarından bir tanesi DDoS (Dağıtılmış Hizmet Reddi) saldırıdır.

Bulut bilişim, yeni nesil veri merkezlerinde kullanılan sanallaştırma teknolojisini kullanarak çok sayıda esnek ve tutarlı hizmet sunan en yeni bilişim teknolojilerinden biri olarak karşımıza çıkmaktadır. Sadece özel şirketler ve bireyler değil, aynı zamanda devlet daireleri de bulut bilişim altyapısı aracılığıyla hizmet kullanılabilirliğini artırmaya çalışmaktadır. Kaynakları yaygın ve şeffaf bir şekilde paylaşma olanağı sağlayan kapasitesi, esnekliği ve maliyeti minimuma indirmesi sayesinde bulut bilişim farklı ihtiyaçları karşılayan prosedürleri uygulayabilme yeteneğine sahiptir. Ayrıca, bulut bilişim kullanıcılara isteğe bağlı hizmetler sunar ve ortak altyapıya erişebilme özelliğine sahiptir.

Bulut sistemler kolay iletişim kanalları sağladığından kurumlarda yapılan işlemlerin verimliliğini ve etkinliğini artırmaktadır. Bu sebeple, bulut bileşenleri ve hizmetleri günümüzde yaygın olarak kullanılmaya başlanmıştır. Bu dijital iletişim metodu, organizasyonların hizmet kalitesi için büyük avantajlara sahip olsa da, birçok güvenlik tehdidine de sebep olabilmektedir. Bulut sistemlerinin karşılaştığı en büyük tehditlerden ve en büyük güvenlik sorunlarından biri hizmet reddi (DoS) saldırıdır. DoS saldırıları, en yaygın web sitelerinin çalışmasını birkaç saat ya da daha fazla süreyle çevrimdışı bırakarak büyük kayıplara ve onarım maliyetlerine neden olmaktadır (Alzahrani et al., 2014). Bu çalışmada bulut sistemlerinde karşılaşılan DDoS saldırıları ve bu saldırıları tespit etmek ve önlemek için kullanılan çeşitli yöntemler analiz edilmiştir.

Bu çalışmada kullanılan kısaltmalar Tablo 1’de belirtilmiştir.

BULUT SİSTEMLERİNE YAPILAN DAĞITILMIŞ HİZMET REDDİ SALDIRILARI VE ALINABİLECEK ÖNLEMLER

Tablo 1: Kısaltmalar Listesi

Kısaltma	Açıklaması
DoS	Hizmet Reddi Saldırıları
DDoS	Dağıtılmış Hizmet Reddi Saldırıları
NIST	Ulusal Standartlar ve Teknoloji Enstitüsü
IaaS	Hizmet Olarak Altyapı
PaaS	Platform Olarak Altyapı
SaaS	Hizmet Olarak Altyapı
HTTP	Hiper Metin Transfer Protokolü
HTTPS	Güvenli Hiper Metin Transfer Protokolü
SOA	Servis Odaklı Mimari
VM	Sanal Makine
GAE	Google Uygulama Motoru
CRM	Müşteri İlişkileri Yönetimi
ERP	Kurumsal Kaynak Planlama
TCP	İletim Kontrol Protokolü
SSL	Güvenli Giriş Katmanı
IDS	Saldırı Tespit Sistemi
IPS	Saldırı Önleme Sistemi
AD	Anormallik (tabanlı) Algılama
SD	İmza (tabanlı) Algılama
IP	İnternet Protokolü
PAD	Vuruş Aritmi Algılama
RED	Rastgele Erken Algılama
RTO	Yedekleme Zamanı Nesnesi
TTL	Yaşam Süresi

2. BULUT BİLİŞİM SİSTEMLERİNİN YAPISI

Bulut bilişim mimarisi, bir ağ üzerinden hizmet olarak sağlanan bilgi işlem kaynaklarının verimli kullanılmasını sağlar. Bulut bilişim hizmetleri, uzak kullanıcı verileri, yazılım ve hesaplama görevlerini yerine getirmekten sorumludur (Mahmood, 2011).

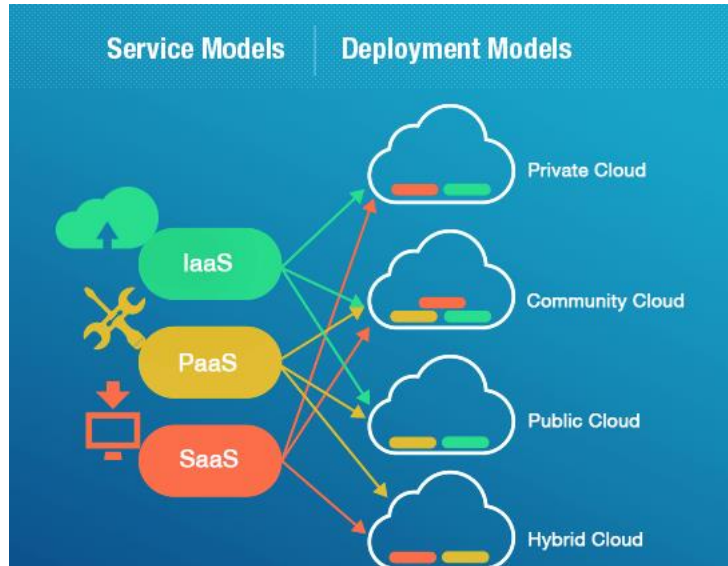
Amerika Birleşik Devletleri Ulusal Standartlar ve Teknoloji Enstitüsü (NIST) bulut bilişimin beş temel özelliğini isteğe bağlı self servis, geniş ağ, erişim kaynak havuzu, ölçülen hizmet ve hızlı esneklik olarak tanımlamaktadır (Mell and Grance, 2011). Ayrıca bulutun karma, topluluk, özel ve genel olmak üzere dört farklı dağıtım modelinde hizmet sunduğunu tanımlar. Bulut sağlayıcıların, hizmetleri hizmet olarak altyapı (IaaS), hizmet olarak platformlar (PaaS) ve Hizmet olarak yazılım (SaaS) olmak üzere üç model halinde sunduğunu belirtir., Şekil 1 bulut hizmet ve dağıtım modelleri ile birlikte bulut ortamının temel özelliklerini göstermektedir.

Bulut hizmet modelleri aşağıdaki gibi özetlenebilir (Fakeeh, 2016):

IaaS - Hizmet olarak Altyapı: Bu katman, hizmet olarak donanımsal bilgi işlem kaynakları sağlar. Örneğin, kullanıcıların uygulamanın altyapı gereksinimlerine bağlı olarak sanallaştırılmış bir sunucuları olacaktır. Kullanıcılar düzenleme katmanı üzerinde kontrol sahibi olmayacak ancak işletim sistemi, dağıtılmış uygulamalar ve depolama üzerinde kontrol sahibi olacaklardır (Ubhale and Sahu, 2013).

PaaS- Hizmet Olarak Platform: Bunlar genellikle hizmet olarak ara katman yazılımı olarak da adlandırılır. Bu düzey, altyapı katmanı üzerinde, .NET, J2EE, vb. programlama dil platformları için kullanıcı desteği sağlayan güvenli bir katmandır. Google App Engine (GAE) veya Microsoft Windows Azure gibi çeşitli ara katman yazılımı hizmetleri bulunmaktadır (Ubhale and Sahu, 2013).

SaaS - Hizmet Olarak Yazılım: Bu bir bulut bilgi işlem ortamının en üst katmanıdır ve temel olarak uygulama barındırma olarak belirtilebilir. CRM, ERP, vb. bir uygulama bulutta barındırılır ve kullanıcılara kullanım bazında veya aylık / yıllık olacak şekilde lisanslanır (Ubhale and Sahu, 2013).



Şekil 1: Bulut Hizmet ve Dağıtım Modelleri

Bulut bilişim ortamında giriş kontrolü, data lokasyonu ve segregasyonu ve araştırma raporları gibi konular kritik güvenlik risklerine sebep olabilmektedir (Zunnurhain ve Vrbsky, 2010) Bulut sistemlerdeki bu tür güvenlik risklerinin azaltılması için bilgi güvenliğinin temel unsurları olan gizlilik, bütünlük, güvenilirlik ve data paylaşımı konularına dikkat edilmesi gerekmektedir. Ayrıca bulut bilişimde yüksek kullanılabilirlik şarttır. Buluttaki kullanılabilirlik, taleplerine göre yetkili kullanıcılar tarafından bulut kaynaklarının ve hizmetlerinin kullanılmasını gerektirir (Varia, 2011). Ancak, veri gizliliği ve hizmet kullanılabilirliğine yapılan saldırılar, kaynak çok yönlülüğü ve

BULUT SİSTEMLERİNE YAPILAN DAĞITILMIŞ HİZMET REDDİ SALDIRILARI VE ALINABİLECEK ÖNLEMLER

paylaşım özellikleri nedeniyle bulut ortamını tehdit edebilir (Oktay ve Sahingoz, 2013). Bulut bilişim ortamında hizmet kullanılabilirliğini tehdit eden en büyük güvenlik saldırılarından biri Dağıtılmış Hizmet Reddi (DDoS) saldırısıdır. Bu saldırı, yasal bulut kullanıcılarının bulut sağlayıcıları tarafından sunulan hizmetlere veya kaynaklara ulaşmasını engeller (Okaför et al., 2016). Bu, sunucunun bilgi işlem kaynaklarının ağ bant genişliğini doldurarak tüketilmesiyle elde edilir ve sonunda bulut hizmetlerinin veya kaynaklarının kullanılamamasına yol açar. Böylece büyük finansal kayıplara neden olur (Somani et al., 2017). Son Arbor Networks güvenlik raporuna göre, bulut bilişim ile ilgili hizmetleri hedefleyen DDoS saldırılarının oranı giderek artmaktadır (Arbor Networks, 2014).

3. BULUT SİSTEMLERİNE YAPILAN SALDIRILAR

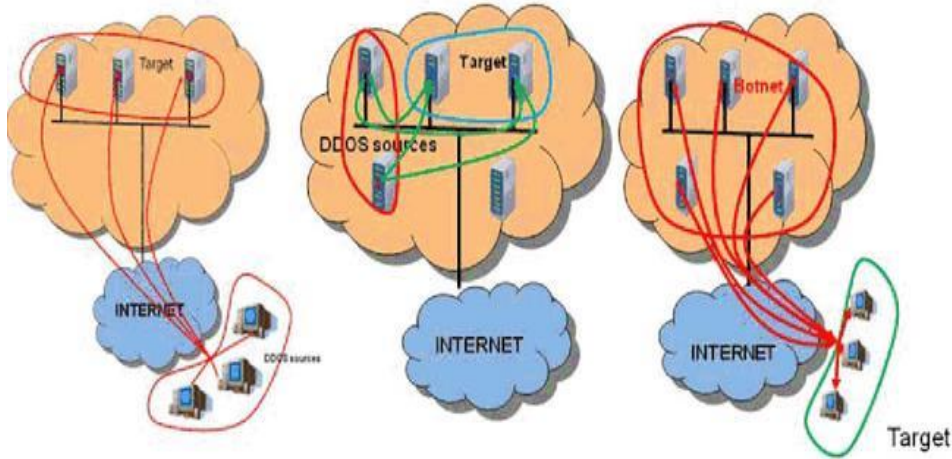
Bulut bilişim sistemlerinin dezavantajı, dışardan gelecek saldırılara karşı savunmasız olmasıdır. Kötü niyetli kullanıcılar sıklıkla hassas veriler elde etmeye veya sistemin işleyişini devre dışı bırakmaya çalışır. Kötü niyetli saldırılar sıklıkla kişisel veya finansal verileri hedef alır. İnternetteki bir siteyi rahatsız etmeyi amaçlayan zararlı bir veri toplama mekanizması tarafından yapılan dijital saldırı, kullanılabilirliğe dayalı saldırı olarak adlandırılır. Bu saldırılar, TCP, HTTP veya HTTPS dalgalanmaları, SSL ve diğer saldırılar gibi çok çeşitli saldırı vektörlerini kullanır. Sonuç olarak, kullanılabilirliği hedef alan saldırılar internet sitelerini etkileyen en önemli güvenlik tehlikeleri arasındadır (King ve Raja, 2013). Bu saldırılara hizmet reddi (DoS) saldırıları denir. Saldırı birden fazla makine tarafından gerçekleştirilirse, dağıtılmış hizmet reddi (DDoS) saldırısı olarak adlandırılır. Hem DoS hem de DDoS saldırılarının sonucunda kötü niyetli saldırganların iyi bilinen ve güvenilir bir sitenin hizmetlerini nasıl kesintiye uğratacağını ya da tamamen durduracağını açıklayan pek çok haber internet ortamında veya medyada yaygın olarak karşımıza çıkmaktadır (Bakshi ve Yogesh, 2010). DDoS saldırısında, genellikle “zombiler” olarak adlandırılan kötü niyetli bir bilgisayar ordusu, bir hedef sunucudan büyük veri toplamak veya sunucuya istek göndermek için birbirleriyle uyumlu şekilde çalışır. Kenardaki ağ düğümlerinde zamanla kaynak sıkıntısı yaşanması olasıdır ve bu düğümler ve giderek daha savunmasız hale gelir. Kaynak taşmaları kenarlara olan mesafe ile ters orantılıdır. Böyle bir etkinin iki nedeni vardır. Sunucuya yakın olan düğüm çoğunlukla ağ kenarına yakın olduğu için daha az hizmet kapasitesine sahiptir. Daha yakın düğümler çoğunlukla daha az kapasiteye sahiptir, bu nedenle daha az kullanıcıyı idare eder. İkincisi, bu tür düğümlerin ağ içinde birleşen saldırılara daha fazla maruz kalması gerekecektir. Aşırı yük durumlarında, tüm sunucu sistemi savunmasız hale gelir ve bozulmaya eğilimlidir. DDoS saldırıları, son on yılda İnternet'e yapılan en büyük ve en yıkıcı saldırı türlerinden biridir (Yau et al., 2005).

Bulut bilişim, temel olarak, özellikle çok sayıda dış ve iç güvenlik sorununa açık olan ve genel bulutları hedef alan sanallaştırma ve Servis Odaklı Mimari (SOA) gibi çeşitli yeni teknolojileri içermektedir (Komu et al., 2012). Bulut ortamında, DDoS saldırılarının, harici Dağıtılmış Hizmet Reddi saldırıları ve dahili Dağıtılmış Hizmet Reddi saldırıları olarak saldırının kökenine bağlı olarak sınıflandırılması mümkündür.

Harici Dağıtılmış Hizmet Reddi saldırıları: harici bir botnet saldırganının bir bulutta çalışan bin veya yüzlerce Sanal Makineyi (VM) kapsayan bir Truva atı başarıyla gönderme ve yükleme yeteneğine sahip olduğu yerlerde ortaya çıkar. Tehlikeye atılan VM'ler veya botnet, harici mağdurlara yönelik başka saldırıların kaynağı olarak kullanılabilir (Latanicki, 2010).

Dahili Dağıtılmış Hizmet Reddi saldırıları: Bu tür saldırılar Harici DDoS saldırılarından daha tehlikelidir. Bulut altyapısının tamamen bozulmasına yol açması mümkündür (Latanicki, 2010). Normalde, bu saldırıda, aynı bulut üzerinde çalışan bir grup sanal makineyi hedefleyen dahili botnet saldırganları bulunur. Sekil 2 bulut altyapısına yönelik dahili ve harici DDoS saldırılarını göstermektedir.

BULUT SİSTEMLERİNE YAPILAN DAĞITILMIŞ HİZMET REDDİ SALDIRILARI VE ALINABİLECEK ÖNLEMLER



Şekil 3: Bulut altyapısına yönelik dahili ve harici DDoS

Bu nedenle, bulutun özellikle de genel bulutların önemli güvenlik açıklarına sahip olması durumunda, bir bulutun birçok harici ve dahili DDoS tehdidinin kaynağı olması mümkündür.

4. BULUT SİSTEMLERİNE YAPILAN DDoS SALDIRILARININ TESPİT EDİLMESİ

DDoS saldırılarının çoğu, hedef ağ, sistem veya web sitesindeki tüm kullanılabilir ağ bant genişliğini veya kaynaklarını tüketerek şekilde tasarlanmıştır. Saldırgan, hedefi kötü amaçlı veya rahatsız edici taleplerle bir engelle doldurmak veya sistemin artık isteklere yanıt vermeyeceği şekilde bir protokolü veya doğal bir güvenlik açıklığını kötüye kullanmak için birçok yöntem ve araçtan birini kullanır. Bir DDoS saldırısının sonucunda, sahte isteklerle uğraşmak zorunda kalan sunucu meşru istek gönderen ve işleminin yapılmasını bekleyen gerçek kullanıcıların isteklerine cevap veremeyecek duruma gelir ve bu durum hizmetlerin aksamasına neden olur. Bu sebeple DoS saldırıları hizmet reddi saldırıları olarak adlandırılır. Tipik bir DoS saldırısının yapısı Şekil 3’de gösterilmiştir.

DoS ve DDoS saldırılarına karşı pek çok yöntem önerilmiştir. Saldırı tespit sistemleri (IDS) bu yöntemlerden bir tanesidir. IDS, şüpheli, anormal veya uygunsuz aktiviteleri tespit eden ve kaydeden donanım ve yazılımları ifade eder. İzinsiz giriş önleme sistemi (IPS), IDS’dekine benzer işlevlere sahiptir, ancak kötü niyetli etkinlikleri önlemek veya azaltmak için gerekli önlemleri alabilmesi bakımından daha karmaşıktır. Çok sayıda çalışma, anormallik tabanlı algılama (AD), imza tabanlı algılama (SD) veya her ikisinin bir melezini içeren IDS tekniklerinden birinin kullanımına odaklanmıştır (Alqahtani et al., 2014). İzinsiz Giriş Önleme Sistemleri, ağ sistemlerini korumak için bir savunma katmanı sunar. IPS, güvenlik duvarı tekniklerini ve saldırı tespit sistemlerini birleştiren proaktif bir ağ tekniği olarak adlandırılabilir. IPS, çeşitli veri kümelerini ve veri kayıtlarını inceleyerek ve saldırı örneği algılama sensörlerini kullanarak saldırıların ağ etkilemesini veya ağa tam olarak girmesini engeller. Bir saldırının başarılı bir şekilde tanımlanması üzerine, Saldırı Önleme Sistemi saldırıyı engeller ve ağır verileri kaydeder. IPS, gelen ve giden verilerin eşleşmesini gerçekleştiren imzalar kullanarak saldırı düzenlerini tanımlar. IPS ayrıca hem gelen hem de giden paketlerin ana bilgisayar algılamasını gerçekleştirir ve herhangi bir tehdit etkinliğini engellemeye çalışır.

Ana bilgisayar tabanlı yaklaşım:

Saldırıları izlemek ve önlemek için ana bilgisayarı veya işletim sistemini kullanır (Ahmed, 2009). Ana bilgisayar tabanlı, IPS için en yaygın standarttır. Ana bilgisayar tabanlı IPS, ana bilgisayar işletim sistemi düzeyinde küçük bir yerleşik program / uygulama yüklenerek dağıtılır. Bu uygulamaya genellikle izleme aracı denir. İzleme araçları, konuşlandırıldığı ana bilgisayardan şüpheli herhangi bir etkinlik olup olmadığını kontrol eder ve merkezi bir izleme istasyonuna bildirir. Bu, IPS’nin saldırıları hedefe ulaşmadan çok iyi önlemesini sağlar. İzleme araçları, etkinlik türüne bağlı olarak bazı uyarılar oluşturur ve özelleştirilebilen önceliklere göre sınıflandırılır. İzleme sistemleri, Dosya

BULUT SİSTEMLERİNE YAPILAN DAĞITILMIŞ HİZMET REDDİ SALDIRILARI VE ALINABİLECEK ÖNLEMLER

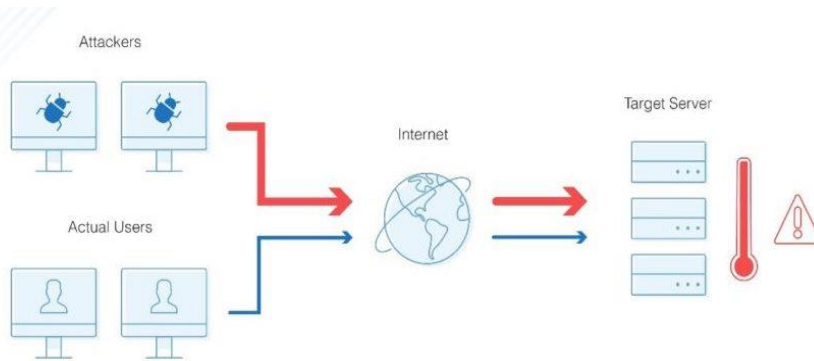
Sistemi İzleme, Günlük Dosyası Analizi, Bağlantı Analizi, Çekirdek Tabanlı Saldırı Tespiti, vb. Tarafından çeşitli düzeylerde izleme yapılır. Sistem ayrıca belleğin belirli parçalarına erişildiğini veya değiştirildiğini kontrol eder ve buna göre hareket eder. İzleme araçları, sistem yapılandırma dosyalarını kontrol eder ve güvenli olmayan ayarlara ve güvenlik ihlallerine meyilli diğer nesnelere tarar. En büyük dezavantaj, kötü niyetli bir saldırganın izleme ajanını değiştirmede başarılı olması durumunda, güvenlik yöneticisi yeterli kontrolleri dağıtmadığı sürece saldırıyı önlemenin bir yolu yoktur.

Ağ tabanlı yaklaşım:

Bu yaklaşım esas olarak ağ üzerinde ağın içinden ve dışından tüm gelen ve giden paketleri inceleyerek, tanımlayarak ve işaret ederek çalışır. Ağ tabanlı yaklaşım, ağa ve sunuculara kapsamlı bir güvenlik sağlamak için diğer ağ tabanlı güvenlik sistemlerinin bir birleşimini gerektirir. Ağ tabanlı yaklaşımın bir avantajı, ana makinelerle herhangi bir istemci veya izleme yazılımının kurulmasına gerek olmamasıdır. “Tuzak”, internet güvenliğinde kullanıcıların saldırganlara karşı masaları açmasını sağlayan yeni bir teknolojidir. Bu teknolojinin, saldırgan hakkında bilgi edinmek amacıyla saldırıya uğraması amaçlanmıştır. IDS'ler, büyük bir saldırı türüne karşı koymak için bilgisayar ağlarında yaygın olarak kullanılır. IDS'nin konuşlandırılması, özellikle çok büyük miktarda gönderilmiş yanlış isteklerin yönetilmesi gibi önemli bir dezavantaj yaratır. Bu dezavantaj, bazı işletme IDS'lerinin günde binlerce istek oluşturabilmesiyle daha da kötüleşir. Bu nedenle, yanlış alarmların azaltılması IDS'lerin gücü ve kullanılabilirliğinde önemli rol oynar (Jensen et al., 2009). IDS saldırı vektörleri ve saldırı alanı hakkında geniş ve derinlemesine bilgi sunar.

Anomali tabanlı Saldırı Tespit sistemi:

Bu sistem sıfır gün saldırılarını gerçek zamanlı olarak önleme yeteneğine sahip olacak şekilde tasarlanmıştır. Anomali tabanlı sistemler, önceden tanımlanmış bir işletim prosedüründen farklı çalışma görüldüğünde anormallikleri veya sapmaları tespit etmeye tasarlanmıştır. Bu anomali tabanlı algılama, bir IDS'nin geliştirilmiş prosedürden sapma yüzdesini hesaplayarak sıfır gün saldırılarını önlemesini sağlar. Hem yerel hem de küresel sıfır gün saldırılarının tespiti bir IDS için en büyük zorluktur. Geliştirilmiş yerel prosedür iyi tanımlandığından, ancak küresel sıfır gün saldırısını tanımlamak hala bir zorluk gibi görüldüğünden, ilkini tespit etmek IDS için oldukça kolaydır. Ancak gerçekte, bu yüksek bir algılama oranına yol açar ve bu da büyük miktarda yanlış alarmlara neden olur. Saldırı, bir sisteme zarar vermeyi veya bir sisteme yetkisiz erişim elde etmeyi amaçlayan kötü amaçlı olaylar veya davranışlar olarak tanımlanır. IDS, bir sistemi tehlikeye atmaya, saldırmaya veya sisteme girmeye çalışan gelen ve giden ağ trafiğini denetleyen yazılımı ifade eder. Bu kurallar farklı senaryolar için değişiklik gösterir. IDS bir hırsız alarmı gibi davranır. IDS izinsiz bir giriş durumunda alarm verir. Bir saldırı başarılı veya başarısız olabilir ve her iki durumda da IDS, gelecekte karşılaştırılabilir tehditlerle başa çıkmak için yönetici tarafından analiz edilebilecek saldırı davranışının kaydedilmesinden sorumludur (Jensen et al., 2009). Saldırı tespiti, imzalı veya anomali tabanlı olabilir (Babaie et al., 2014).



Sekil 3: Tipik Bir DoS Saldırısı

5. BULUT SİSTEMLERİNE YAPILAN DDOS SALDIRILARININ ÖNLENMESİNDE KULLANILAN BAZI YÖNTEMLER

Bu bölümde bulut bilişim sistemlerine yapılan DDoS saldırılarını önlemek için kullanılan yöntemlerden bazıları analiz edilmiştir.

Modellenen Saldırı Algılama (MAD) ve Periyodik Saldırı Algılama (PAD) Yöntemleri:

Sahte atlamalı sayım yöntemi, sahte paketleri saptamak ve ardından atmak için kullanılır (Jin et al., 2003). Paketin hedefine ulaşmak için atlacağı bazı sıçrama noktalarına işaret ettiği için paketin TTL değerini değiştirmek mümkün değildir. IP başlığının yaşam süresi, atlama sayısı bilgisi oluşturmak için kullanılır. Burada Hop sayımı filtreleme yöntemi, hop sayımlarına dayalı olarak tüm IP adreslerini tarayarak bir hop atlamalı sayım eşleme tablosu oluşturmak için kullanılır. Oluşturulan eşleme tablosu, sahte ve aynı olan IP paketlerini algılar. Atlama sayısı mekanizması, paket varış zamanını bulmak için bulanık mantık tekniği kullanılarak daha doğru yapılabilir. Modellenen saldırı algılama yöntemi zaman serileri üzerinde çalışır ve PAD tekniği yakın periyodik trafik doğasına bağlı olduğu için periyodik saldırı tespit yönteminden çok kısa sürede saldırıları tespit edebilmesi için örnekleme yapar (Subhashini et al., 2012).

Dinamik Algılama Yöntemi:

Dinamik algılama yönteminde kurbanın konumuna yakın olan kenar yönlendiricilere yerleştirilir. Bu tür yönlendiriciler, mağdur ağlara daha da bağlı olan yönlendiricilerin çıkış bağlantı noktalarına saldırı algılaması yapacaktır. Düşük hızda iletim kontrol protokolü saldırısı algılanırsa, yönlendiricinin saldırı trafiğinin alındığından giriş portlarını doğrulaması gerekir. Algılama daha sonra etkilenen yönlendiricinin tüm giriş bağlantı noktalarında gerçekleştirilir. Giriş bağlantı noktasında düşük oranlı bir saldırı algılanırsa, etkilenen yönlendirici algılamayı giriş bağlantı noktasına bağlı olan yukarı akış yönlendiricilerinin herhangi birine veya tümüne engeller. Etkilenen yönlendirici, giriş bağlantı noktalarından herhangi birinde düşük hızlı bir saldırı bulamazsa, bu düşük hızlı saldırının aşırı dağıtılmış bir şekilde dağıtıldığı anlamına gelir. Düşük oranlı bir saldırı tespit edildiğinde, bu geri itme mekanizması, saldırının potansiyel olarak saldırı kaynağına yakın olduğunu tespit etmek için kullanılır. Geri itme mekanizması, etkilenen iletim kontrol protokolü akış miktarını en aza indirir (Thatte et al., 2005).

Akıllı Gönderme Oranları ve Arabellek Boyutu:

TCP akışı, hedef yönlendiricinin tampon depolama kapasitesini artırarak sivri saldırılara karşı koruma sağlayabilir. Akış yönetimi ve kuyruk yönetimi yüksek bağlantı kullanımı elde etmek için kullanıldığında, TCP akışı [21] 'de belirtildiği gibi sivri ataklar kullanılarak korunabilir. Rastgele erken algılama şeması olarak yönlendirilen düşük DoS saldırıları algılanmaz, yönlendiricideki tıkanıklığı önlemeye yardımcı olur, bu da daha uzun kuyruk boyutu ve yüksek patlama hızları dikkate alır. Saldırının uzun tamponu doldurmak için yüksek hızlarda ilemesi gerekir. Bu nedenle bunlar düşük oranlı saldırılar değildir ve RED algoritması ile tespit edilemez (Sarat ve Terzis, 2005).

Rastgele Yeniden İletim Zaman Aşımı:

Sistemlerin çoğunda 1 saniyelik yeniden iletim zaman aşımı (RTO) vardır. Bu standart değer, düşük oranlı DOS saldırıları tarafından kullanılmaktadır. RTO belirlenen herhangi bir değere ayarlanırsa, saldırının doğru RTO'yu algılayamayacağına inanılır. Bu parametre kullanılarak saldırı hızı kontrol edilebilir. Akışlar düzgün bir şekilde izlendiğinde ve RTO randomizasyonu kullanılarak, paketlerle ilgili sorunlar tespit edilebilir ve geriye doğru izleme saldırının kağıtta açıklandığı gibi bulunmasına yardımcı olur [22]. Bu, saldırıları tespit etmenin etkili bir yolu değildir. Çünkü bu yöntem düşük oranlı DDOS saldırılarını algılayamaz. Saldırılar meşru görüldüğünde, bu teknik tamamen başarısız olur ve bu tür saldırıları tespit edemez veya önleyemez (Yang et al., 2004).

Paket Yüzdesi ve Kuyruk Ön Bellek Yöntemi:

Bu yöntemde hedef yönlendiricinin önbellek kuyruğu araştırılır. Algılama yöntemi, gelen trafiği yönlendiriciden gelen trafik akışına göre düzenler. Bu, aktif kuyruk yönetimi sağlar [23]. Burada, düşük DDoS saldırılarını tespit etmek için ağırlıklı boğulma ile anormalliği durdurmak kullanılır. Önerilen HAWK yöntemi, Hizmet Reddi trafik akışını tespit etmek için düşürme algoritmaları kullanır. Uyarlanabilir olmayan trafik akışı ve uyarlanabilir trafik akışı arasında adalet sağlar. Saldırı tespiti iki parametre referans alınarak gerçekleştirilir. Birincisi, önbellek sırasını hedef yönlendiricideki paketlerin yüzdesi için analiz etmektir. TCP paketinin saldırgan akış yüzdesi doğrulanarak yapılır. İkinci parametre eşik yüzdesidir, Bazı istemciler ve saldırganların eşik yüzdesi kullanılarak hesaplanır. TCP akışı, eşik değerini kullanarak düşük DOS saldırısını algılar (Kwok et al., 2005).

E-Firecol:

Bu yöntem, İnternet Servis Sağlayıcısı (ISS) seviyesine yerleştirilecek saldırı önleme sisteminden oluşur. Bu İzinsiz Girişi Önleme Sistemi, hizmet reddi saldırılarına karşı savunmak için ana bilgisayarlar etrafında bir sanal koruma halkası oluşturur. Ayrıca, seçilen trafik bilgilerinin değişimi ile işbirliği yapılır. Ancak ISS düzeyinde koruma, web hizmetlerine saldırı yapılmayacağını garanti etmez. Dolayısıyla bu sistem yavaş ve düşük uygulama seviyeli DDoS saldırılarına karşı hala savunmasızdır. Saldırı Önleme Sistemi geleneksel olarak bir güvenlik duvarı kullanarak yapmıştı. Gerçek kullanıcıların durum bilgileri durum bilgisi olan güvenlik duvarları kullanılarak korunabilir. Ancak bunlar yüksek hacimli DDoS saldırılarını önleyemez (Chandra et al., 2013). Bu güvenlik duvarları ağ katmanında en iyi şekilde çalışır, ancak uygulama katmanı saldırılarında olduğu gibi düşük ve yavaş saldırıları algılayamazlar.

Çok Düzeyli DDOS Saldırı Algılama:

Bu yöntem mevcut anomali tabanlı algılama sisteminin entropi tabanlı sistemlerle birleştirilmesiyle sağlanır. İlk düzeyde, kullanıcılar, birçok algılama algoritması olan bir yönlendiriciden geçiş isteğinde bulunarak meşru istekler için ağ düzeyinde doğrulanır. İkinci düzeyde, bulut ortamındaki başka bir yönlendirici saldırıları algılar ve eşiklerden geçirilir. Bağlantı bir eşik değerinin üzerindeyse, meşru kabul edilir. Yoksa bir saldırı olarak algılanır. Üçüncü taraf, bu sistemi temsil etmek ve sürdürmek için kullanılır. Bir saldırı algılandığında kullanıcıyı bilgilendirilir (Navaz et al., 2013). Yönlendiriciler ağ bağlantılarıyla ilgileneneği ve uygulama katmanı saldırıları istek düzeyinde hedefleneceğinden, bu yaklaşım istekler için geçerli değildir. Eşik değerleri, saldırının yüksek sel saldırısı veya düşük saldırı olup olmadığını yeterince algılamaz ve sistemin meşru kullanıcıyı engellemediğine dair net bir açıklama yoktur.

Akış hızı filtresi:

Bu filtreyle düşük oranlı DDOS saldırıları tespit edilir. Trafik hızı kademeli olarak artar ve ağ ana bilgisayarlarına düşük oranlı DDOS saldırıları tarafından saldırı yapılır. DDOS saldırıları akış seviyesi filtresi ile engellenir (Antikainen et al., 2014). Bulut bilişim, talep üzerine farklı hizmetler sunar. Yazılım bulmacası, son kullanıcıya, kullanıcının erişim elde etmeden önce doğru şekilde çözmesi gereken bir yazılım bulmacası ile sağlanacağı bir teknolojidir. Kullanıcı buluttaki servislere erişim talep eder ve servis sağlayıcı veya sunucu bir bulmacayla yanıt verir. Bulmacanın başarıyla çözülmesi erişime yol açacaktır. Kullanıcı yanlış bulmaca erişimi sağlarsa engellenir. DDOS saldırıları için yazılım veya makineler kullanılarak çok sayıda istek gönderilir. Burada, eşik tabanlı istek yöntemi DDOS saldırılarını çözmek için kullanılır. Kullanıcının belirli bir zamanda yalnızca belirli sayıda istekte bulunmasına izin verileceği anlamına gelir. Eşik değerini geçen herhangi bir kullanıcı, sistem tarafından engellenecektir (Subramaniam et al., 2016).

Sinir Ağları ve Veri Madenciliği Yöntemi:

DDoS saldırılarını tespit etmek için kullanılan yöntemlerden bir diğeridir. Bu model daha az belleğe ihtiyaç duyar ve daha hızlı algılandıklarını iddia eder. Sonuç, TCP saldırılarının çoğunun tespit edildiğini göstermektedir (Shamsolmoali et al., 2014). Bu sistem katman 7 (Uygulama Katmanı)

BULUT SİSTEMLERİNE YAPILAN DAĞITILMIŞ HİZMET REDDİ SALDIRILARI VE ALINABİLECEK ÖNLEMLER

saldırıları tespit etmede yardımcı olmaz ve çok fazla yük taşır. TTL değeri yöntemi, düşük ve yavaş saldırıları tespit etmede yardımcı olmaz. TTL değeri için bulutta paket izlemenin, buluttaki DDoS saldırılarını tespit etmede daha büyük avantajları olduğu söylenir, ancak bulut hizmet sağlayıcıları için bir sınırlama oluşturan sistem performansını yavaşlatır (Sattar et al., 2015). Yüksek akım DDoS saldırılarının çoğu TTL değeri izleme kullanılarak algılanabilir, ancak düşük ve yavaş saldırılar gizli kalır ve sistem bu tür saldırılara karşı oldukça savunmasız olma eğilimindedir.

Donanım Tabanlı Filigran Teknolojisi:

DDoS saldırılarına karşı etkili bir savunma yapmanın zorluğundan biri, saldırıya uğrayan trafiği yasal trafikten ayrı olarak tanımlamaktır. Birçok sahte adres, saldırganlar tarafından sisteme saldırmak için kullanılır ve böylece her veri paketini kontrol etmek için kaynak tüketir. Ters kontrol mekanizması yardımıyla, veri paketinin kaynağını işaretlemek için kenar yönlendiriciler kullanılır. DDoS saldırısı durumunda belirli alanlardan büyük miktarda veri gelir. Kaynak IP oluşturulduysa, çoğu durumda verilerin türü aynı olacaktır. TTL veya hop sayımlarının yardımıyla veri paketleri güvenilir veya güvenilmez olarak sınıflandırılır. Bu işlemi gerçekleştirmek için “donanım tabanlı filigran teknolojisi” kullanılabilir (Rahman et al., 2014).

Honeypots:

Uygulama katmanındaki DDoS saldırıları, web seliyle belirli bir hizmeti hedeflemeye çalışır. Örneğin, HTTP sel saldırıları, sunucu kaynaklarını etkilemek amacıyla bir sunucuya yüksek oranda otantik uygulama katmanı isteği gönderir. DDoS saldırıları daha az bant genişliği tüketir ve tanınması çok daha zordur, çünkü saldırgan kimlik doğrulama istekleri nedeniyle sunucuya saldırır (Wong et al., 2014). DDoS saldırılarıyla başa çıkmak için çeşitli yöntemler önerilmiştir, ancak her seferinde değişikliklere saldırganın bir yolu vardır, hiçbir öneri DDoS saldırılarını tamamen engellemez. Honeypot bu yaklaşımlardan biridir. Bunlar, bir ağda bilgisayar korsanlarının denemelerini incelemek ve ağ yöneticilerini olası bir saldırı hakkında uyararak için kullanılan izlenen kodlara benzer. Öncelikle bilinen DDoS ve gelecekteki varyantlara karşı yüksek olasılıkla işletim ağımızı savunmamız gerekmektedir. İkinci olarak, uzlaşmanın kaydedilmesi saldırganlara karşı yapılacak yasal bir işlemde yardımcı olabileceği için saldırganın yakalanması sağlanabilir.

6. SONUÇ

DDoS saldırıları günümüzde bilgi güvenliği teknolojileri için büyük bir tehdit oluşturmaktadır. Bu saldırılar özellikle de bulut hizmetlerinin kullanılabilirliğini önlemeye yönelik çalışmaktadır. Gün geçtikçe yeni ve geliştirilmiş bir saldırı türünün ortaya çıkmasıyla birlikte DDoS saldırılarına karşı geliştirilen savunma mekanizmaları da etkinliğini kaybetmektedir. Aslında DDoS saldırılarına karşı koruma sağlayan bu savunma mekanizmaları her zaman tek başına etkili değildir. Özellikle hibrit savunma mekanizmaları oluşturmak için farklı mekanizmaları birleştirmek ve farklı bulut bilgi işlem katmanları ile çalışmak gerekmektedir. Bu sebeple farklı DDoS saldırılarının bulut sistemi üzerindeki etkilerini araştırmak son derece önemlidir.

Bu çalışmada, farklı DDoS saldırılarının bulut ortamına etkisi araştırılmıştır ve bu saldırıları engelleme amacıyla yapılan farklı araştırma çalışmaları analiz edilmiştir. Bulut sistemlerindeki olası DDoS saldırıları ve bunları tespit etmek ve önlemek için kullanılan yöntemlerin bazıları listelenmiştir. Bulut için DDoS saldırılarını tanımlamak adına oldukça fazla araştırma yapıldığı halde henüz saldırılar için mükemmel bir çözüm üretildiği söylenemez. Kullanılan saldırı tespit ve önleme yöntemleri geliştirilmeye açıktır. Seçilen savunma yöntemleri ise saldırının türüne ve içeriğine göre farklılık göstermektedir. Farklı türdeki DDoS saldırılarının önlenmesi ve tespiti için yeni ve modern yöntemler geliştirilmek zorundadır. Mevcut ve gelecek çözümlerin karşılaştırması ile çeşitli DDoS saldırılarını azaltmak için değerlendirme metrikleri ile bulut tabanlı deneyler gerçekleştirilebilir. Son olarak bulut hizmet sağlayıcılarının gelecekteki DDoS saldırılarını tespit etmek ve azaltmak için ortak çalışmaları incelenebilir.

7. KAYNAKÇA

Ahmed, M., Pal, R., Hossain, M. M., Bikas, M. A. N., & Ruhunabi, A. B. M. (2009, April). PIDS: A packet based approach to network intrusion detection and prevention. In *2009 International Conference on Information Management and Engineering* (pp. 124-127). IEEE.

Alzaharani, A., Alalwan, N., & Sarrab, M. (2014, April). Mobile cloud computing: advantage, disadvantage and open challenge. In *Proceedings of the 7th Euro American Conference on Telematics and Information Systems* (pp. 1-4).

Arbor Networks, "10th Annual worldwide InfrastructureReport".<http://pages.arbornetworks.com/rs/arbor/images/WISR2014.pdf>, 2014.

Babaie, T., Chawla, S., & Ardon, S. (2014). Network traffic decomposition for anomaly detection. *arXiv preprint arXiv:1403.0157*.

Mell, P., & Grance, T. (2011). The NIST definition of cloud computing.

Fakeeh, K. A., & King, J. (2016). An Overview of DDOS Attacks Detection and Prevention in the Cloud. *network, 11*(7).

Ubhale Mr. P. R & Sahu, Prof. A. M. "Securing Cloud Computing Environment using Intrusion Detection and Prevention System (IDPS),"International Journal of Computer Science and Management Research Vol 2 Issue 5 May 2013.

Zunnurhain, K., & Vrbsky, S. (2010, December). Security attacks and solutions in clouds. In *Proceedings of the 1st international conference on cloud computing* (pp. 145-156).

Oktay, U., & Sahingoz, O. K. (2013, September). Attack types and intrusion detection systems in cloud computing. In *Proceedings of the 6th International Information Security & Cryptology Conference* (pp. 71-76).

Varia, J. (2011). Best practices in architecting cloud applications in the AWS cloud. In *Cloud Computing: Principles and Paradigms* (Vol. 18, pp. 459-490). New Jersey, USA: Wiley Press.

Okafor, K. C. (2016). Vulenarability Bandwith Depletion Attack on Distributed Cloud Computing Network.

Somani, G., Gaur, M. S., Sanghi, D., Conti, M., & Buyya, R. (2017). DDoS attacks in cloud computing: Issues, taxonomy, and future directions. *Computer Communications, 107*, 30-48.

King, N. J., & Raja, V. T. (2013). What do They Really Know about Me in the Cloud: A Comparative Law Perspective on Protecting Privacy and Security of Sensitive Consumer Data. *Am. Bus. LJ, 50*, 413.

Bakshi, A., & Dujodwala, Y. B. (2010, February). Securing cloud from ddos attacks using intrusion detection system in virtual machine. In *2010 Second International Conference on Communication Software and Networks* (pp. 260-264). IEEE.

Yau, D. K., Lui, J. C., Liang, F., & Yam, Y. (2005). Defending against distributed denial-of-service attacks with max-min fair server-centric router throttles. *IEEE/ACM Transactions on Networking, 13*(1), 29-42.

Latanicki, J., Massonet, P., Naqvi, S., Rochwerger, B., & Villari, M. (2010, April). Scalable Cloud Defenses for Detection, Analysis and Mitigation of DDoS Attacks. In *Future Internet Assembly* (pp. 127-137).

Komu, M., Sethi, M., Mallavarapu, R., Oirola, H., Khan, R., & Tarkoma, S. (2012, September). Secure networking for virtual machines in the cloud. In *2012 IEEE International Conference on Cluster Computing Workshops* (pp. 88-96). IEEE.

BULUT SİSTEMLERİNE YAPILAN DAĞITILMIS HİZMET REDDİ SALDIRILARI VE ALINABİLECEK ÖNLEMLER

Alqahtani, S. M., Al Balushi, M., & John, R. (2014, March). An intelligent intrusion detection system for cloud computing (SIDSCC). In *2014 International Conference on Computational Science and Computational Intelligence* (Vol. 2, pp. 135-141). IEEE.

Jensen, M., Schwenk, J., Gruschka, N., & Iacono, L. L. (2009, September). On technical security issues in cloud computing. In *2009 IEEE International Conference on Cloud Computing* (pp. 109-116). Ieee.

Mahmood, Z. (2011, August). Cloud computing: Characteristics and deployment approaches. In *2011 IEEE 11th International Conference on Computer and Information Technology* (pp. 121-126). IEEE.

Jin, C., Wang, H., & Shin, K. G. (2003, October). Hop-count filtering: an effective defense against spoofed DDoS traffic. In *Proceedings of the 10th ACM conference on Computer and communications security* (pp. 30-41).

Rahman, M., & Cheung, W. M. (2014). A novel cloud computing security model to detect and prevent DoS and DDoS attack. *International Journal of Advanced Computer Science and Applications*, 5(6).

Subhashini, K., & Subbalakshmi, G. (2012). 'Tracing sources of DDoS attacks in IP networks using machine learning automatic defence system. *International. Journal. Electron. Commun. Comput. Eng.*, 3, 164-169.

Thatte, G., Mitra, U., & Heidemann, J. (2008, April). Detection of low-rate attacks in computer networks. In *IEEE INFOCOM Workshops 2008* (pp. 1-6). IEEE.

Sarat, S., & Terzis, A. (2005, October). On the effect of router buffer sizes on low-rate denial of service attacks. In *Proceedings. 14th International Conference on Computer Communications and Networks, 2005. ICCCN 2005.* (pp. 281-286). IEEE.

Yang, G., Gerla, M., & Sanadidi, M. Y. (2004, June). Defense against low-rate TCP-targeted denial-of-service attacks. In *Proceedings. ISCC 2004. Ninth International Symposium on Computers And Communications (IEEE Cat. No. 04TH8769)*(Vol. 1, pp. 345-350). IEEE.

Kwok, Y. K., Tripathi, R., Chen, Y., & Hwang, K. (2005, August). HAWK: Halting anomalies with weighted choking to rescue well-behaved TCP sessions from shrew DDoS attacks. In *International Conference on Networking and Mobile Computing*(pp. 423-432). Springer, Berlin, Heidelberg.

Chandra, R., & Gudavalli, M. (2013). E-FireCol to Detect Multiple DDOS Attacks. *International Journal of Advanced Research in Computer Science and Software Engineering*, 3(12).

Navaz, A. S., Sangeetha, V., & Prabhadevi, C. (2013). Entropy based anomaly detection system to prevent DDoS attacks in cloud. *arXiv preprint arXiv:1308.6745*.

Antikainen, M., Aura, T., & Särelä, M. (2013). Denial-of-service attacks in bloom-filter-based forwarding. *IEEE/ACM Transactions On Networking*, 22(5), 1463-1476.

Subramaniam, T., & Bethany, D. (2016). Preventing Distributed Denial of Service Attacks in Cloud Environments. *International Journal of Information Technology, Control and Automation*, 6(2).

Shamsolmoali, P., Alam, M. A., & Biswas, R. (2014). C2DF: High rate DDOS filtering method in cloud computing. *International Journal of Computer Network and Information Security*, 6(9), 43.

Sattar, I., Shahid, M., & Abbas, Y. (2015). A review of techniques to detect and prevent distributed denial of service (DDoS) attack in cloud computing environment. *International Journal of Computer Applications*, 115(8).

BULUT SİSTEMLERİNE YAPILAN DAĞITILMIŞ HİZMET REDDİ SALDIRILARI VE ALINABİLECEK ÖNLEMLER

Wong, F., & Tan, C. X. (2014). A survey of trends in massive DDoS attacks and cloud-based mitigations. *International Journal of Network Security & Its Applications*, 6(3), 57.